

Automatic Exam Marking (AEM) Pipeline for GCSE-level Assessments

Candidate Number: 249694

BSc Computer Science

Department of Informatics

Project Supervisor: Hsi-Ming Ho

Word Count: 11,754

Statement of Originality

"This report is submitted as part requirement for the degree of Computer Science BSc at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged. I hereby give / withhold permission for a copy of this report to be loaned out to students in future years (delete as necessary)."

A handwritten signature in black ink, appearing to be 'J. Mayhew', written on a light-colored, textured background.

Signature

Table of Contents

Glossary	VI
1 Introduction	1
1.1 Project Overview	1
1.2 Related Works	1
1.2.1 LLM-Based Auto-grading	1
1.2.2 “Towards Automated Evaluation of Handwritten As- sessments” (Rowtula et al., 2019)	3
1.2.3 BERT Family of Models	4
2 Problem Area	8
2.1 Manual Marking	8
2.2 Current Attitudes of AEM in Education	8
2.3 GCSE Exams	10
3 Development - Overview	14
3.1 Preface	14
3.2 Requirements Analysis	14
3.2.1 Functional Requirements	16
3.2.2 Functional Requirements	18
3.2.3 Domain Requirements	19
3.3 Tools	19
3.3.1 Python	19
3.3.2 ChatGPT API	20

3.3.3	Graves' Handwriting Synthesis Model	20
3.4	Data	23
3.4.1	SciQ Dataset	23
3.4.2	RiceChem Dataset	24
4	Development - Synthetic Paper Generation	25
4.1	QAQ Exams	25
4.1.1	Considerations for Parsing	25
4.1.2	Sequencing	27
4.1.3	Multiple Choice Question (MCQ)	28
4.1.4	Short-form Questions (SF)	29
4.1.5	Long-form Questions (LF)	29
4.2	Data Extraction	30
4.2.1	Pre-Processing	33
4.2.2	Mark Scheme Interpretation	34
4.3	Calligrapher	36
4.4	Question Paper Generator	39
4.4.1	Textual question (0)	39
4.4.2	Multiple Choice Question (1)	40
4.4.3	Text Block (2)	40
4.5	Mark Scheme Generator	41
5	Development - Page Parsing	44
5.1	Page Parsing	44

5.1.1	Alternative Approaches	44
5.1.2	Question Paper Parser	45
5.1.3	Mark Scheme Parser	48
5.2	Parsing Effectiveness	49
6	Development - Evaluator Module	51
6.1	8.1 Reasoning Models and Chain of Thought	51
6.2	Evaluation Algorithm	53
6.3	8.3 Prompt Engineering	55
6.4	Cost/Efficiency	56
7	Development - Application/GUI	58
7.1	User Interface	58
7.2	Requirements Fulfillment	59
7.3	Requirement Fulfillment	60
8	Experiment: Final Performance Evaluation of AEM Pipeline	68
9	Discussion	78
9.1	Findings	78
9.2	Future Research	79
9.3	Limitations	80
10	Conclusion	81
	References	81

A Appendix	92
-------------------	-----------

Appendix	93
-----------------	-----------

Glossary

AEM Automatic Exam Marking

MCQ Multiple Choice Question

SF Short-form Question

LF Long-form Question

QPP Question Paper Parsing Module

QPG Question Paper Generator

MSP Mark Scheme Parsing Module

MSG Mark Scheme Generator

EM Evaluator Module

DG Data Generation Module

EG Exam Generator

CAL Calligrapher

LLM Large Language Model

DNN Deep Neural Network

MVP Minimum Viable Product

1 Introduction

1.1 Project Overview

In this project, we propose an end-to-end marking solution intended for streamlining in-classroom mock GCSE examinations in order to explore if using Large Language Models in automated exam marking could relieve teachers of current assignment marking strains within their work.

We aim to explore why AEM integration within education has not accelerated despite mounting pressure from increasing teacher workload (**Green, Felstead, Gallie and Henseke, 2018**), with particular focus on assignment marking as to how current procedures in educational examination could be changed in order to accommodate intelligent LLM integration.

Due to the sensitive nature of the data with which we would be working to create such a project, a major part of development will be to generate an arbitrarily large set of completed exam scripts. The insights gained from building such a system will be used to explore the financial and operational implications of the integration of AEM solutions.

1.2 Related Works

Throughout this project, we have had the opportunity to research autograding in real depth, concluding that there has been little academic exploration of the applications of LLMs in the automatic assignment grading for real-world use by teachers/examiners (**Alkafaween, 2025**)(**Chu et al., 2025**)(**Ito and Ma, 2025**).

In this section, we will recap some of the foundational research upon which this project is based on, as well as take a look at some newer approaches that have been proposed since.

1.2.1 LLM-Based Auto-grading

A paper that explored independent LLM question grading was done by Flodén (2024), in which 463 answers from the University of Gothenburg Logistics and

Transport Masters’ Course were sampled and manually input 3 times into ChatGPT, the score response for each was then averaged out to give the final score. The following prompt was used:

“I want you to act as a university professor grading a written exam.
Please score the following answer on a scale of 0 to X. Question:
QQQ. Answer: AAA.”

This approach is relatively naive, and assumes “Black box” behaviour (Sam, Finzi and Kolter, 2025) when dealing with the model as an evaluator. This approach provides no additional marking guidance, and is likely to mark based on syntactic correctness and pre-trained data assumptions already contained within the model.

In the end, it was found that this approach produced predictions within 5% of the human marker grade 32.4% of the time.

Morjaria et al (2024) mark long-form medical exam papers by providing them as input for ChatGPT, alongside a rubric, a grading standard, both the rubric and the grading standard, or neither.

The study finds that the model tends to inflate grades, and display peak performance when no rubric is provided. This is likely due to the fact that the total concatenated length of the question, student essay, and mark schemes each marking input can range into the thousands of tokens, far exceeding the models optimal context window leading to attention dilution (Qin et al 2024). The LLM’s inability to effectively distribute attention to key parts of the text would certainly impair its ability to reason for presence of key mark scheme concepts, especially in such a technically dense domain as Medicine.

It is important that when proceeding with LLM grading that mark assignment decisions are made using tangible rubric information, and efforts are made to accommodate to the strengths and weaknesses of these models when doing so (Fan et al., 2025).

An approach that was not covered in the initial proposal for this project was done by Ito and Ma paper (2025), who used a multi-step Ensemble ToT LLM marking process each LLMs marking habits are first studied through a “Pseudo-learning” phase, having those LLMs then mark question independently, and finally entering a moderated “debate” phase in which the

LLMs collectively decide on a final mark while factoring in their marking biases learnt in Pseudo-learning.

This approach displayed a label accuracy of 77.87%, an improvement **Ito and Ma (2025)** claim to be 6.97% over other state of the art approaches. Though this approach is a noticeable improvement in performance and follows recent trends in consensus mechanism integration to boost LLM performance (**Chen et al., 2025**), the computational cost of using such an approach would be unlikely to be worth the trade-off in accuracy.

1.2.2 “Towards Automated Evaluation of Handwritten Assessments” (Rowtula et al., 2019)

This solution marks handwritten answers using “Word Spotting”, a technique in which the words in a Textual Reference Answer (TRA) are used to create an expanded dictionary of relevant keywords, after which a handwriting synthesiser is used to generate handwritten versions of each token. These word images are matched to segmented word images within the handwritten test based on visual similarity, with the number of matches being a predictor of the student’s grade.

Though an interesting approach to handwritten question marking, it has little practical use to us as it merely finds the semantic similarity between two texts, with one of them being handwritten. This could of course be done with a mark scheme instead of a TRA but would, like in the original paper, only showcase keyword recall and not mark scheme concept application.

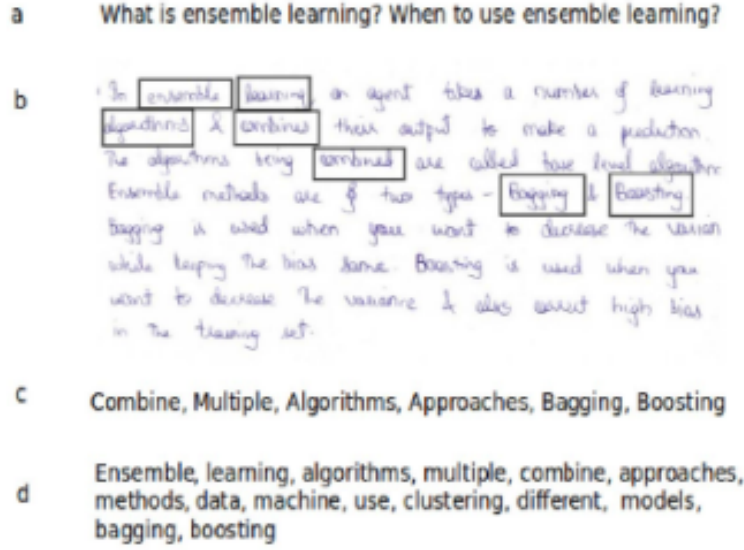


Figure 1: Diagram of parsing mechanism from (Rowtula et al., 2019)

1.2.3 BERT Family of Models

Many recent approaches have leveraged custom deep neural networks with BERT embeddings for automated exam scoring (Sharma, Jayagopi 2024)(Li et al 2023). BERT embeddings capture low-level syntactic as well as high-level semantic features of the text. Google’s RoBERTa-large model, though being outperformed by ChatGPT at its release in sentiment analysis as well acceptability (writing quality), was able to perform better a majority of inference tasks (Zhong et al., 2023).

Since then, performance of both architectures on the same NLE and inference benchmarks (CoLa, SST2, MRPC, STS-B, QQP, MNLI, QNLI, RTE) seen in (Zhong et al., 2023) have be re-evaluated by (Ito and Ma, 2025), who finds the RoBERTa-large to achieve significantly higher correlations (above .0.90 Pearson’s and Spearman’s Corr, ChatGPT at P Corr 80.9, S Corr 72.4) in text similarity tasks (STS-B) as well as a consistent 0.1–2 difference in paraphrase identification.

Model	MRPC		QQP	
	<i>Accuracy</i>	<i>F1</i>	<i>Accuracy</i>	<i>F1</i>
BERT-base	90.0	89.8	80.0	80.0
RoBERTa-large	92.0	92.0	90.0	89.4
ChatGPT	66.0	72.1	78.0	79.3

Model	STS-B	
	<i>Pearson's Corr</i>	<i>Spearman's Corr</i>
BERT-base	83.0	81.9
RoBERTa-large	92.9	91.1
ChatGPT	80.9	72.4

Figure 2: roBERTa-large paraphrase performance compared to ChatGPT in Zhong et al., 2023

In this same study, ChatGPT is also shown to have dominant performance in Logical Inference (QNLI, MNLI and RTE) metrics across the board. This shows that given a mark scheme section (premise), Chat, as well as other LLMs, would be more suitable in confirming whether a student answer (hypothesis) “entails” the mark scheme, largely out-weighing its lack of performance in less relevant metrics.

Model	MNLI			RTE	
	<i>Entailment</i>	<i>Contradiction</i>	<i>Neutral</i>	<i>Entailment</i>	<i>Not Entailment</i>
BERT-base	88.0	88.0	72.0	76.0	64.0
RoBERTa-large	84.0	92.0	88.0	92.0	76.0
ChatGPT	92.0	96.0	90.0	96.0	80.0

Figure 3: roBERTa-large inference performance compared to ChatGPT in Zhong et al., 2023

This project aimed to take findings within these papers further by also incorporating a reasoning models, which should provide new insights into LLM developments in inference.

3.2.4 Competitor Analysis

Graide

Graide is currently the most successful and robust solution resembling our tool. Graide promises to make 89% cuts (from an assumed 11.2 to 1.2 minutes) to teacher marking time by providing bespoke tests and AI marking. They offer 3 main types of questions: Multiple Choice, Short Form and Long

Form. Unlike our proposed solution, they have their own platform and are not intended to mark any pre-existing exams or problem sets.

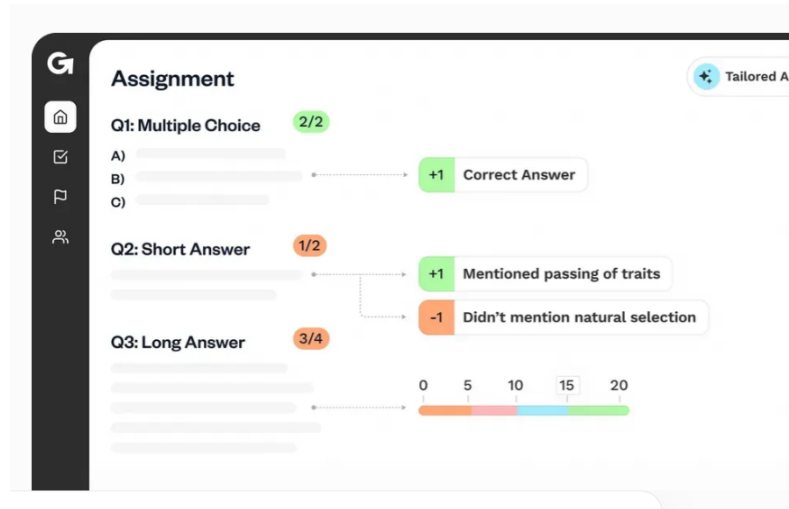


Figure 4: Graide.ai promotional screenshot

GradeWrite.ai

An online essay grading tool with a 10,000 word limit, and a maximum of 30 documents. The platform allows for implementation of custom rubrics, but does not have handwriting OCR capabilities.

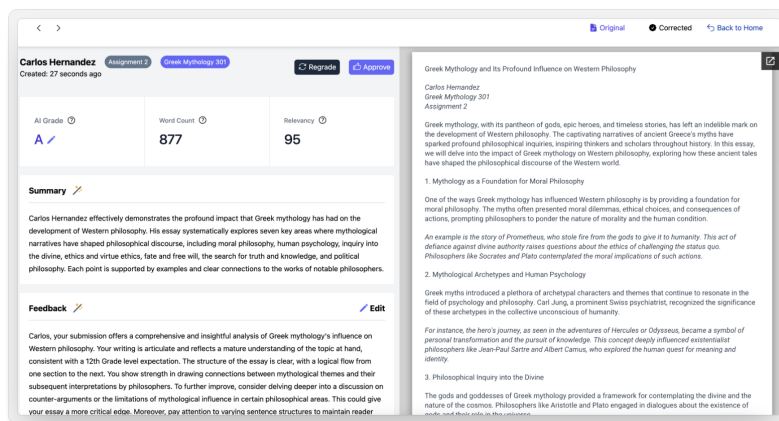


Figure 5: GradWrite.ai usage screenshot

TopMarks.ai

This service provides handwriting parsing, and promises performance “bet-

ter than a human analyser”. TopMarks boasts a 1.5 average marks spread around the Ground Truth against the 10 compared to typical LLMs. Their website serves more as a base for teachers to acquire text-only problems to give out to the class rather than mark already completed ones not present on the website.

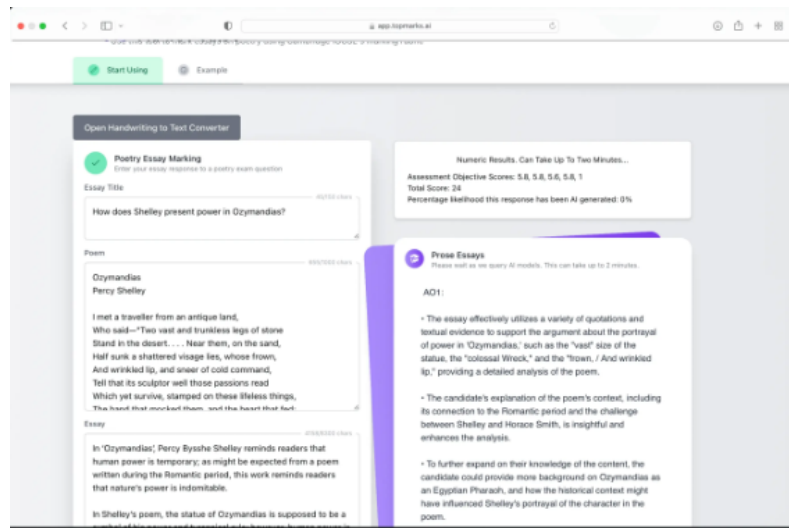


Figure 6: TopMarks.ai usage screenshot

Professional Considerations

This project has adhered to professional and ethical guidance outlined in the **BCS Code of Conduct (BCS, 2022)**. No human test participants were involved during the development of this project, ensuring there to be no impact on individual privacy, health, well-being or the environment (**Section 2.1.1**).

Professional Integrity has been rigorously upheld during the development and reporting of this project (**2.1.2**) (**2.1.4**). No Academic Integrity or University of Sussex Informatics and Engineering department (our relevant authority) rules were broken in the development and reporting of this project. This project has been entirely in the interest of gaining formative insights (**2.1.3**).

The **User Compliance Form**, as the only ethical measure deemed necessary by my supervisor, and has been included in the Appendix.

2 Problem Area

2.1 Manual Marking

Of the professions sampled by Work Intensity in Britain Survey (**Green et al., 2018**), teachers are classified as the highest stressed out of any field, with 4 in 10 being labeled as being under “High Strain”, the highest classification within the study, which is more than double the 17% figure for all other workers. According to (**Morris et al., 2024**) teachers in the UK spend on average 48.76 hours a week working with 9.05 ($\sim 18.5\%$) of that time being spent on marking.

It has been proven that the emotionally and physically taxing nature of grading assignments can increase marking bias and decrease accuracy (**Henderson-Brooks, 2021**)(**Fischer, 2024**), with these effects worsening under more stressful conditions.

The in a government report 2023 (**Department for Education and The Rt Hon Gillian Keegan, 2023**) a 6% increase in FTE (Full-time Equivalent teachers) is said to be observed since 2010/2011, but the report fails to mention that in the same period participating student numbers have risen by 12% (**Department for Education, 2024**)(**Full Fact, 2023**). This concentration of teacher responsibility not only leads to decreased learning standards, but also job conditions which have been proved to be a predictor of attrition amongst qualified teachers (**Dupriez, Delvaux and Lothaire, 2016**).

Due to this it is imperative we look to automation solutions in order to alleviate teacher administrative load (**Demszky et al., 2025**) and allow students to receive prompt and accurate assessment feedback.

2.2 Current Attitudes of AEM in Education

While teaching professionals have been adopting in-classroom solutions to automatically track student in-classroom performance, they have typically been incorporated in low-stakes segments of education (**Demszky et al., 2025**) such as assignments, homework and in-class performance analytics.

On the other hand, automating government-run, final year exams is a contentious issue. The practice is extremely distrusted by teachers due to the lack of accountability and scale of consequences in the case that a student will be incorrectly marked (**Bergviken Rensfeldt and Rahm, 2023**).

For this project, we anonymously surveyed the teachers at The Charter School North Dulwich, a London secondary school, regarding their marking habits as well as attitudes towards automated marking solutions. The subjects taught by the teachers spanned English, Mathematics, Science, Business, Compute Science, and Economics

In this survey, 80% of the teachers expressed being in some way uncomfortable with AEM integration, with none being fully receptive to the idea. They also expressed little interest in the marking of subjective works such as essays, with the “inability to apply the mark scheme accurately” being a concern.

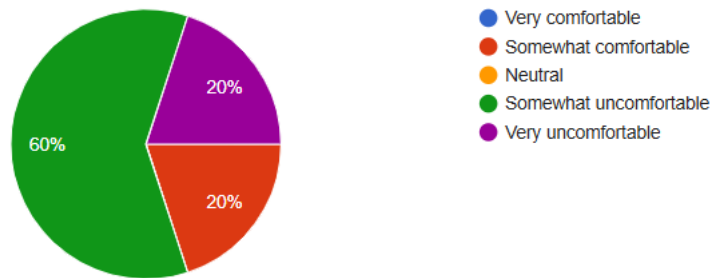


Figure 7: Pie Chart of results from the Charter survey for the question “How comfortable are you with integrating automation into classroom marking?”

Due to this, we will be looking to target the automatic marking of formative, in-classroom mock examinations that are given in preparation for national examination periods. These filled in exam papers for which still have to be hand-marked by teachers, which still create large amounts of workload during said periods.

The adoption of such a solution in this lower stakes environment could provide the foundation for integration on the wider institutional level.

2.3 GCSE Exams

There are 3 main exam boards that deal with the national GCSE examinations: AQA, OCR, and Edexcel.

Exams are typically formatted as a series of mixed-form questions with an exam length of about 1–2 hours and 100 marks. Design practices between exam board and subject are relatively consistent due to stringent standardisation with the goal of maximising student comprehension (**Fowler, 2024**).

17

2.2 Define the term 'social cost'. (2 marks)

2.3 Using Item 8 and Figure 9, calculate the estimated total cost per kilometre of HS2. Give your answer to the nearest £. (5 marks)

Answer: £

2.4 Analyse the possible impact(s) upon labour markets of the cancellation of part of Phase 2b of HS2 to Leeds. (5 marks)

Turn over ►

18

Extra space

2.5 Explain two possible external benefits of HS2. (5 marks)

1.

2.

Figure 8: Two sequential pages from the AQA Economics June GCSE Paper 1. The first page shows 2 Short Form (SF) textual questions as well as one Long Form (LF) question that overflows lines onto the next page. The constituent parts of each textual question in parsing order (up-to-down and left-to-right) are Question Number, Question, Mark, Answer, with the first 3 elements being a reliable delimiter between questions.

Do not
write
here

0 1 . 2 Which species of bird in Figure 1 do scientists think are most closely related? **[1 mark]**

Tick (✓) **one** box.

Brambling and chaffinch	☐
Brambling and goldfinch	☐
Bullfinch and chaffinch	☐
Bullfinch and goldfinch	☐

0 1 . 3 Scientists think the brambling and the bullfinch belong to different species.

What evidence is used by scientists to classify the brambling and the bullfinch as different species? **[1 mark]**

Tick (✓) **one** box.

The brambling and the bullfinch are different sizes.	☐
The brambling and the bullfinch cannot breed together to give fertile offspring.	☐
The brambling and the bullfinch live in different parts of the habitat.	☐
The brambling eats mainly seeds and the bullfinch eats mainly insects.	☐

Question 1 continues on the next page

Turn over ►

Figure 9: Two sequential Multiple Choice Questions (MCQ) from the AQA Biology 2023 GCSE Exam 2 Question Paper. Question Number, Question, Mark delimiter hypothesis applies.

Question	Answers	Extra information	Mark	AO / Spec Ref.
01.7	any two from: - to check toxicity - to check dosage - to check its efficacy	allow to check it is safe allow to check for side effects allow to check it is not poisonous / dangerous / harmful allow to check how much is needed allow to check it works allow to check it does not interact with other drugs	2	AO1 4.3.1.9
01.8	(writers / companies may get) financial gain or (competitor may suffer) financial loss	ignore because the report has not been peer reviewed ignore have not used a double-blind trial	1	AO3 4.3.1.9
01.9	have the claims peer reviewed		1	AO1 4.3.1.9
Total Question 1			13	

Figure 10: Page from the AQA Biology 2023 GCSE Exam 1 Mark Scheme. Table headers are Question, pertaining to question number, Answer, pertaining to hard marking guidance, Extra Information, pertaining to edge case handling in marking, Mark, pertaining to the maximum amount of mark allocatable, and AO/Spec. Ref referencing Assessment Objective applied.

J 277/02			Mark Scheme		J une 2023
Question			Answer	Mark	Guidance
			- will complete a final iteration once sorted (to check for no swaps needed) - moves items up the array - end of array becomes sorted first - moves/bubbles the highest value to the top - less efficient/slower than insertion sort (on large sets of values) ... more iterations / comparisons (on average) ... when data more scrambled		
3	(c)	(ii)	1 mark each to max 2 e.g. - Both produce a sorted list / array - Both work in place / without duplicating data / without using divide and conquer - Both need a temporary variable - Both use loops / iteration / repeats - Both loops are nested / inside each other - Both (may) need multiple passes - Both use selection - Both work with an array / list data structure - Both work from left to right - Both build up sorted list one item at a time (after every pass) - Both compare (pairs of) values - Both (typically) less efficient / slower than merge sort (or other sorting algorithms)	2 (A01 1b)	Allow reference to both sorting / putting items into order for BP1. "Allows sorting of numbers and strings" meets BP1 Allow answers relating to not needing additional memory as BP2. Allow "breaking into smaller lists" as divide and conquer for BP2. If answer is a statement (e.g. "uses loops"), assume candidate is talking about both algorithms doing this.

14

Figure 11: OCR Computer Science 2 June Exam Mark Scheme. It can be seen to contain a Question column, Answer, providing direct guidance on how to mark the question, mark assignment per question, as well as an (additional) Guidance column providing marking edge case handling

To test the effectiveness of our software it was imperative to procure a sufficient testing dataset. Correspondence with AQA representatives as well as the Project Supervisor highlighted the difficulty of acquiring filled in exams scripts from examining bodies.

Additionally the provision of completed GCSE exam scripts would violate the Data Protection Act 2018 Section 25.2 (**UK Government, 2018**) due to information determining a student's exam mark being considered personal data, and would only be accessible with the express permission of each student.

The sourcing of a sufficiently large number of authentic, full-length exam scripts via administering in-house mock examinations would likely require the involvement of students or individuals within the target key-stage. Factoring in the ethical overhead created by proceeding with this approach, as well as the relatively small amount of data that could be created this way, it did not seem like a viable course of action for this project.

Hence, we have chosen to use publicly available and as well as data we've been granted access to procedurally synthesise the completed handwritten exam scripts we require.

3 Development - Overview

3.1 Preface

In the interest of integrity and the provision of overarching project planning vision it is important to address the changes made to the approach outlined in the Interim report “*Automatic Exam Marking (AEM) Pipeline for GCSE-level Assessments*”.

Development Paradigm: In the initial report it was outlined that the project will be taking a Waterfall design approach (**Senarath, 2021**) in order to reach its goal of delivering the solution. Due to the highly iterative approach the development of this project in practice and exploration of the interplay of the between the Procedural Exam Generator, Question Paper and Mark Scheme Parsers, and Evaluator modules, it would be more fitting to describe the approach taken as AGILE (**Cramer et al., 1997**). Hence it would be more appropriate to describe the target outcome of the project to be an MVP (**Lenarduzzi and Taibi, 2016**) and the insights that come from its development.

Technologies: In the Interim Report, it was planned that Java was going to be used in the development of the GUI as well as the “file management system” due to potential performance gains via JIT optimization (**Tudor and Walter, 2006**). The expedition of the GUI to Java was seen to create needless technical overhead which was seen to not outweigh the development time costs. In its stead, the PyQt5 (**Willman, 2021**) library was used to implement the GUI to the same effect. Different options could be explored if future development if we looked to explore various optimisations of the solution for scaling across many users and institutions. The file management system also proved to not require any extra work, and an option for the user to customise export options was added to the final software instead.

3.2 Requirements Analysis

During the aforementioned Charter School North Dulwich survey, we asked teachers certain questions to gain insights into the needs for various features for our software. We also gave the teachers the opportunity to provide open-ended suggestions for our solutions.

Through the survey we gained the following insights: - The ability to handle multiple types of questions (MCQs, SF, LF) was crucial to 80% of the surveyed teachers. Flexibility in swapping out markschemes was also important to almost all teachers.

- The ability to scale the function of the software to the classroom was crucial, with it being important or extremely important to all teachers

- The main value-add of an AEM solution was time saved, with multiple teachers independently stating that it would help relieve them of high marking workload. Several

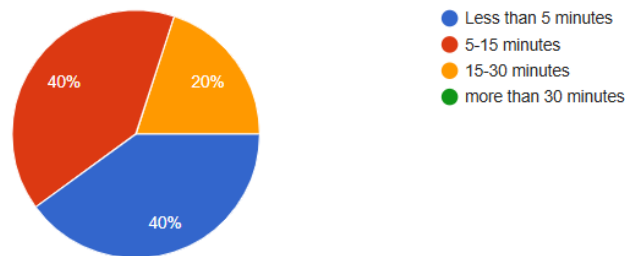


Figure 12: Self-reported “Average time spent marking a returned student assignment”, with an average of 9.5 minutes.

- As mentioned before, teachers were extremely untrusting of AEM software marking integrity. Questions regarding parsing and evaluation methods arose, with transparency in evaluation being paramount.

- Functions that were of little importance: Integration with Learning Management Systems (LMS), Accessibility for all users, automatically re-marking tests if mark-schemes were updated.

3.2.1 Functional Requirements

The following are the exact requirements outlined in the Interim Report at the outset of the project's development.

Requirement Description	Input	Output	Fulfilment Status
Upload in Mark Scheme section opens file Explorer for Markscheme	Upload button pressed	File explorer opened	Requirement Satisfied
Indication of Successful upload	One PDF file selected and uploaded	File name displayed with tick under file selection to indicate success	Requirement Satisfied
No invalid uploads allowed	More than one PDF or non-PDF selected	Error indicating condition not satisfied	Requirement Satisfied
Upload in Answer Section Opens files in explorer for answers	Upload button pressed	File explorer opened	Requirement Satisfied
Successful PDF upload	One PDF selected	PDF added to scrollable container	Requirement Satisfied
Batch upload	≤ 30 PDFs selected	PDFs added to scrollable container	Requirement Satisfied
No invalid uploads	>30 PDFs or non-PDF files selected	Error indicating condition not satisfied	Requirement Satisfied
Directory upload	Directory with only PDFs selected	PDFs added to scrollable container	Requirement Satisfied
Uploaded pages visualised	File(s) uploaded	Display first page of first document	Requirement Satisfied
Choice in visualised page	File selected in scrollable container	Display selected page	Requirement Satisfied

Requirement Description	Input	Output	Fulfilment Status
Visualiser blank on open	Programme started	Display nothing	Requirement Satisfied
Page navigation work within page range	Navigation button pressed within range	Show next/previous page in range	Requirement Satisfied
No navigation outside of page range	Navigation button pressed outside range	Do nothing	Requirement Satisfied
Invalid start	“Start” pressed with inputs but no config	Output only final mark	Requirement Satisfied
“Show Marks Per Question” scan runs successfully	Option selected, Start pressed	Per-question results shown	Requirement Satisfied
“Chain of Reasoning” scan launches successfully	Option selected, Start pressed	Reasoning outputs shown	Requirement Satisfied
Target directory selection gated by export checkbox	Export selected	Directory picker becomes active	Requirement Satisfied
No export if no target directory set	Export selected, Start pressed	Error shown requesting directory	Requirement Satisfied
Can successfully export results as Excel	Export and directory selected, Start pressed	Excel results saved	Requirement Satisfied

3.2.2 Functional Requirements

The following are the exact requirements outlined in the Interim Report at the outset of the project's development.

Statement	Requirement	Description	Fulfilment Status
Teachers took 9.5 minutes on average per student	Speed	Software shall mark each test in a timespan no longer than 2 minutes	Requirement Satisfied
Scalability as an important factor	Scalability	Software will be able to handle a batch of exams up to 30 at once	Requirement Satisfied
Teachers concerned over incorrect marking	Reliability	Evaluator will reach at least 85% accuracy with human evaluators	Requirement Satisfied
Teachers are not comfortable with new software	Accessibility	UI will be made to obfuscate the complexity of LLM and only allow basic function	Requirement Satisfied
Software should get better over time	Modularity	Software should be easy to maintain when better models are developed in order to easily improve software performance and reliability	Requirement Satisfied

3.2.3 Domain Requirements

Statement	Requirement	Description	Fulfilment Status
Software intended for teachers	Usability	Software should represent the software that might already be found within education, such as Microsoft Office	Requirement Satisfied
Work for a standardised set of exams	Compatibility with QAQ Exam papers and mark scheme	Should be able to parse answers and layout effectively given a QAQ exam	Requirement Satisfied
Ability to handle multiple types of questions	Layout Parser	Layout Parser will be trained on both multiple choice but also text questions	Requirement Satisfied
Lack of Accountability	Accountable	Chain of Reasoning used to evaluate LLM marking, increase performance, and when needed returned as output	Requirement Satisfied
Do not create classroom implementation costs that exceed the usefulness of the tool	Efficiency	Large batch sizes with marking times significantly shorter than ones done by hand	Requirement Satisfied

3.3 Tools

3.3.1 Python

Python is an interpreted high-level general purpose language most commonly known for its simple syntax and robustness when it comes to applications in automation and machine learning (**Yarlagadda, 2018**).

The main tasks within our project will be to use libraries to algorithmically generate PDF, process and output comma-separated data, make calls to the OpenAI API, and create synthetic handwriting. We will not be needing access to lower level hardware function or memory space so we will not be requiring any language beyond Python for this project.

3.3.2 ChatGPT API

The OpenAI API offers programmatic access (**Auger and Saroyan, 2024**) to an array of LLM models with capabilities across nearly all generative and analysis tasks. API costs for the project were self-funded, with one of the metrics for the success being the ability to minimise compute wastage (**Shekhar et al., 2024**).

For this project we will be using 4o-mini and o1-mini models for QPP, MSP and evaluation.

One of the downsides of offloading computation to LLMs are various security risks (**Kumar and Ahmed, 2024**), especially with public institutions such as hospitals, government facilities and in our case, schools and exam boards. These institutions store and deal with extremely sensitive data, have high potential impact of breach (**Rodrigues et al., 2024**). Institutions are hence are a frequent targets of cyber-attacks. One approach for organisations to gain access to LLMs but remain secure has been to self-host LLMs on local servers and use Retrieval-Augmented Generation (RAG) (**Sun et al., 2024**), in order to safely enhance outputs with live information. Our aim to reduce compute expenses will still remains relevant even when models are accessed this way.

3.3.3 Graves’ Handwriting Synthesis Model

During initial project supervisor discussion, the use of a “handwriting font” to the completed exam script data was suggested. Though this would have undoubtedly saved a significant portion of time for other parts of the project, it was essential to the creation of believable exam scripts was the need for handwriting generation that reflected the randomness, internal handwriting consistency, and potential for human error of real human students.

Picking how the handwriting generation was going to be implemented was one of the hardest parts of the project. In recent years there has been almost no developments in the field of handwriting synthesis in the sense that is relevant to this project (**Díaz et al., 2025**). Recent papers and available projects focus on handwriting copying, style augmentation or altering existing databases of handwritten content.

The model we picked for handwriting generation in this project was the Vasquez implementation **(2018)** of a model based on the Graves **(2013)** paper. The paper outlines a Recurrent Neural Net (RNN) with Long Short-Term Memory (LSTM) **(Al-Selwi et al., 2024)** initially trained for next-stroke prediction in incomplete handwriting sequences.

RNNs are DNN architectures which are able to take variable-length sequential inputs by feeding weight-adjusted outputs from previous steps back into subsequent ones **(Sherstinsky, 2020)**. One problem with this architecture is that it may lead to the exploding gradient issues **(Ribeiro et al., 2020)**, in which the learning process is destabilised through gradient accumulation (particularly in extended input sequences), a process in which the looped value grows or shrinks exponentially with a repeatedly looped value and prevent the arrival to a local minima during training.

LSTM networks are a specialisation of RNNs that use an input gate, forget gate and an output gate, along with a long-term and short-term memory states **(Qin et al., 2023)**. The Long-term memory state is weighted based on current short term and input values at the forget gate in order to determine how much of it is kept for the current cell. Then, at the input gate, a percentage of the Potential Long-term memory is added to the Long-term memory state, which is then used to calculate the short term memory state for the next cell at the output gate.

This allows the network to inform outputs based on all previous states within a given input sequence without destabilising gradient descent as outlined before.

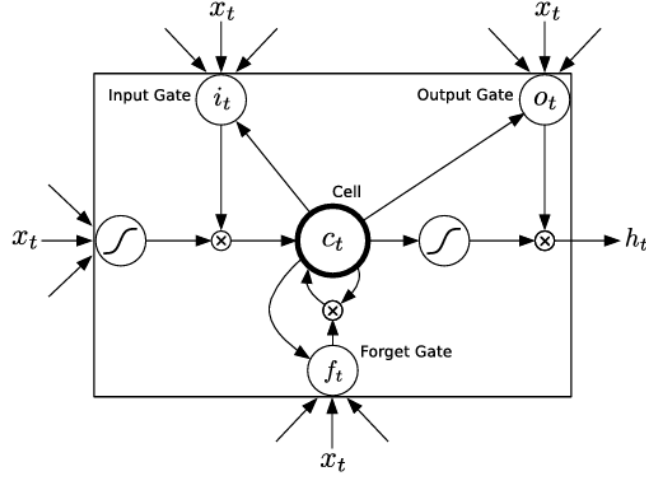


Figure 13: Diagram of LSTM cell from the Graves Model

The output layer of the model predicts the parameters for a mixture of Bivariate Gaussians (Asharabi and Alhazmi, 2024) which are calculated one timestamp at a time based on the trajectory of the previous stroke. Simultaneously, the network tries to predict if the upcoming stroke is set to be the end of a line using a logistic sigmoid function (Geng et al., 2024).

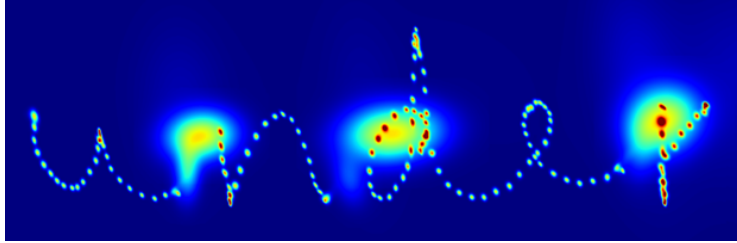


Figure 14: Mixture density outputs for handwriting prediction from the Graves paper

Above is an example in which the word “under” is written, with the bivariate gaussian distributions having wider spread close towards the end of a letter, and the pen is seen to leave the page and with the return location is unknown.

In order to have the network create handwritten text instead of predicting the next stroke, a new model is incorporated that normalises all of the bivariate

Gaussian component outputs, denoted by weight (π), mean (μ), standard deviation (σ), and correlation (ρ), and samples areas with the highest likelihood to be representative of the next stroke.

As this sequence is being generated, the synthesis network implores a window layer (part of the synthesis network), to which the initial string to be written is presented, and is used by the network to understand what part of the string it is currently writing.

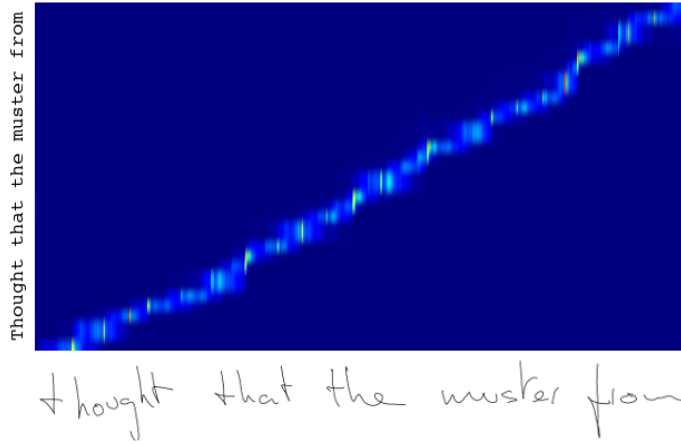


Figure 15: Window weights during a handwriting synthesis sequence

The outputs of the next-stroke-prediction model are hence used as the inputs for the synthesis model with a delay in order to prevent cyclic dependencies within the RNN. The generation of the text using the human data-trained model along with this delay creates the impression of the text being handwritten in real time.

3.4 Data

3.4.1 SciQ Dataset

The SciQ dataset is publicly available and was used by **Welbl et al. (2017)**, with 13,679 science questions spanning physics, chemistry, and biology. Each row consists of the question, the correct answer and 3 “distractor” answers. These could be turned into both the MCQs (by randomly setting the correct answer to A–D) as well as the SF answers (by handwriting the correct answer

text when we want to have the answer be correct, and the distractor otherwise).

3.4.2 RiceChem Dataset

This dataset was used in (Sonkar et al., 2024) and contains 1264 long-form student answers spanning 4 chemistry questions. Along with each question comes a mark scheme or rubric. Each “student answer” row also comes with information on which marks from the rubric it qualifies for.

The use of the dataset was approved by Prof. Shashank Sonkar at Rice University, Houston and was sent to me for the project on 24th Oct 2024.

416347: In each statement below (a-c), two observations are given which seem to contrast with each other. Using your knowledge of electron configurations, orbitals, Coulomb's law, and/or atomic and molecular structures, briefly explain why both of these observations are true, and how the two observations can be reconciled in each case.
 a) b) If light is used to excite an electron to a higher energy level in an atom, only certain frequencies of light can be absorbed. However, if it is used to eject an electron from the atom, any value above a minimum threshold frequency can be absorbed. What's up with that?
 This question can be answered reasonably in around 150 words or fewer.

Figure 16: RiceChem Question 2 “Question”

Correctly states that frequency is proportional to energy of light
 Explaining sentence 1: energy levels of an electron in an atom are quantized
 Explaining sentence 1: FULLY explains energy/frequency absorbed must equal the difference in energy levels in an electron
 Explaining sentence 2: a minimum amount of energy is needed to eject an electron
 Explaining sentence 2: any additional energy becomes kinetic energy

Figure 17: RiceChem Question 2 Mark scheme (Rubric)

The fact that only certain frequencies of light can be absorbed in order to eject an electron to a higher energy level suggests that electrons only exist in certain energy levels or electron shells. Once removed from the electron (ie. absorb light at or above its minimum threshold frequency), however, electrons can exist at any energy level as they are no longer bound or affected by the positively charged nucleus of the atom they were once apart of A

Figure 18: RiceChem Question 2 Student Answer c9f19d62-f8e2-4b34-912e-ed1454bca6ec

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
SD	Score	Correctly states	Explaining sentence	Explaining sentence	Explaining sentence	Explaining sentence	Explaining sentence	Incorrect statement	Incorrect	Blank	Adjustment	Comments		
c9f19d62-f8e2-4b34-912e-ed1454bca6ec	2	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE				
153fbae-72c5-4ee8-b229-07b	1	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE				
1048d3a-1165-4391-9a05-4d	6	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE				
e146e45-6548-4032-993c-7c	8	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE		Amazing job!		
b8b8422-a15c-4421-b530-11	3	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE				
93c2514-9c25-442b-a4a0-9c	7	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE				
04003ad3-04cc-4af4-8af7-5af	4	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE				
9cac451-197c-429c-8e82-af	5	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE				
a079a8b-5277-4c32-b26a-04	8	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE				
f34a079a-409b-409f-416b-04	9	EAI Q2	EAI Q2	TDI Q2	EAI Q2	EAI Q2	EAI Q2	EAI Q2	EAI Q2	EAI Q2				

Figure 19: RiceChem Question 2 Rubric c9f19d62-f8e2-4b34-912e-ed1454bca6ec

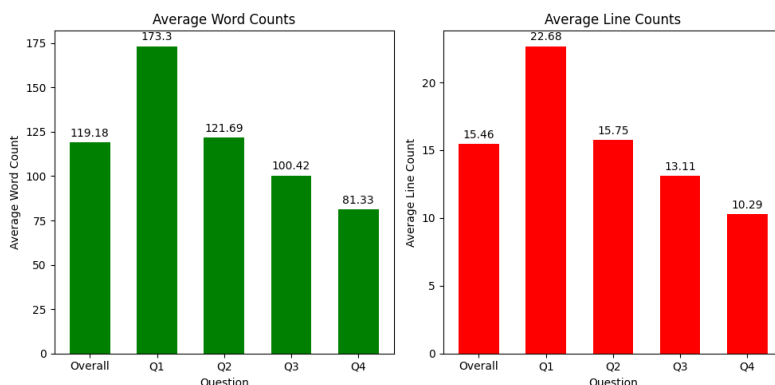


Figure 20: The average word and line counts per question in the RiceChem dataset. The lines were counted by wrapping questions to the last word before the 50 char limit used when using the handwriting model.

4 Development - Synthetic Paper Generation

4.1 QAQ Exams

In order to test the effectiveness of our software we must create a way to generate completed exam scripts based on ground truth data that we could then measure the success of the parsing/evaluation with.

To create exams for our own exam board, “QAQ”, we first look at practices implored by the existing boards.

4.1.1 Considerations for Parsing

Whether parsing pages using CV (**Voulodimos et al., 2018**) or some newer LLM Vision (**Mulfari et al., 2016**) approaches, similar factors can be mitigated in order to increase parsing success.

As peripheral features such as the front-covers, preambles, and various in-page elements such as barcodes will be discarded after parsing and are purely cosmetic (do not affect student marking process), we will be omitting them in our data generation process.

Places where text and non-text elements are mixed would need to be iden-

tified by the parsing model and then parsed separately. High text density as well as proximity of text and non-text objects are extremely likely to interfere in parsing (Zhang et al., 2024).

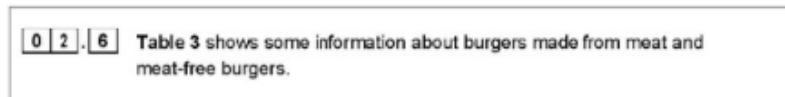


Figure 21: Part of a question header from an AQA Biology exam paper

[0[2].[6] Table 3 shows some information about burgers made from meat and
box
meat-free burgers.

Figure 22: Tesseract image to string parse of the same region

0 2 . 6 Table 3 shows some information about burgers made from meat and
meat-free burgers.

Figure 23: Question number edited to only include textual elements

0 2.6 Table 3 shows some information about burgers made from meat and
meat-free burgers.

Figure 24: Tesseract image to string parse of improved region

For our own implementation we also removed all

Figure 25: An MCQ from the AQA exam board which uses tick boxes for student answer selection

In the QAQ exam board features Question Numbers of varying depth to test parsing correctness but did not incorporate the former mechanisms, having question content order and test-area interrelation be completely random as we do not have to worry about question sequencing effects on human candidates (**Paretta and Chadwick, 1975**). Given the number of questions (amount of labels needed) in the exam, a label sequence is created with randomly seeded depth to which questions would be assigned randomly.

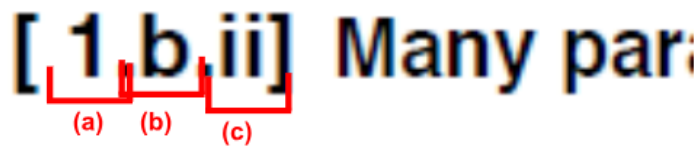


Figure 27: Structure of a Question Number, consistent throughout QAQ papers.

- (a) Top-level label, integer in range (1:)
- (b) Mid-level label (alphabetical), optional, can only increment if the Top-level label didn't last question
- (c) Bottom-level label (lowercase roman numeral), optional, can only exist if mid-level present, can only increment if Mid-level did not increment last level

4.1.3 Multiple Choice Question (MCQ)

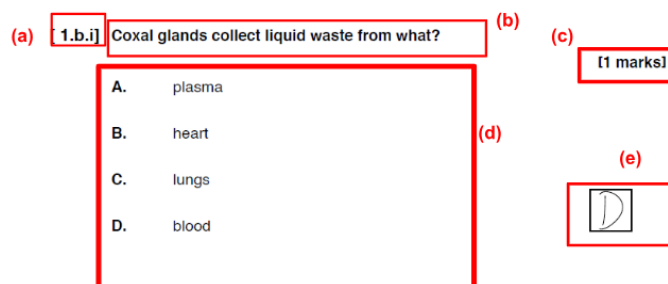


Figure 28: An example of a filled in QAQ MCQ

All text is Helvetica at 10 units tall:

- (a) Question Number

- (b) Question, always on the same level as Question Number
- (c) Max Mark, slightly below the Question
- (d) Choices directly below the Question
- (e) Answer Box and Student Answer below and left of Max Mark

4.1.4 Short-form Questions (SF)

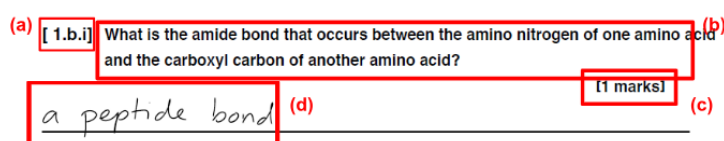


Figure 29: Structure of a SF Question

- (a) Question Number
- (b) Question
- (c) Max Marks
- (d) Student Answer (Text Block), line 415 units wide. Handwriting is scaled to the more limiting of the width of the line and height of 25 units

4.1.5 Long-form Questions (LF)

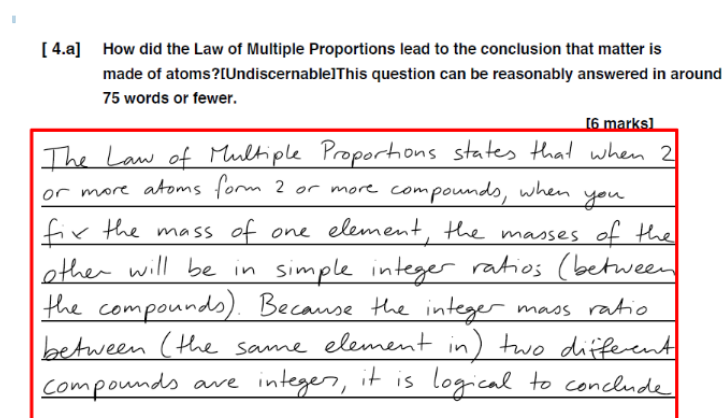


Figure 30: Structure of a Longform Question

(Highlight) Text Block behaves similarly to SF but line overflows at the end of the word before 50 char. limit.

The ground truth Student Answer for each LF is stored as plaintext, which is then wrapped to the needed line length (≤ 50 chars). The visual representation of the answer will scale exactly to the amount of wrapped lines, with no lines being left blank thereafter.

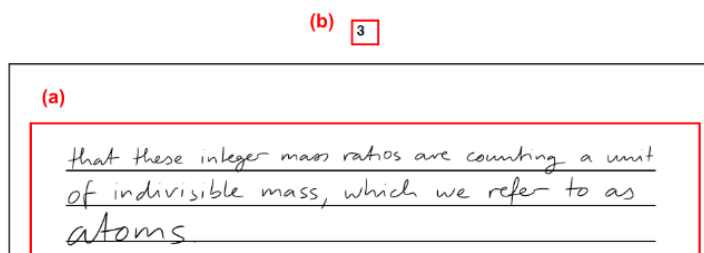


Figure 31: LF Overflow

- (a) Textblock can only exist alone on a page only if an LF overflows onto the next page. This only happens if it can write at least 2 lines before running out of space. If there isn't enough space for 2 lines, the entire question is moved onto the next page.
- (b) Page number at page top, sequential

4.2 Data Extraction

To create a QP-MS pair, symmetric data must be generated from our datasets and be fed as input into our Question Paper Generator (QPG) and Mark Scheme Generator (MSG) algorithms. This ensures both are consistent to each other and the synthesised MS can be used to mark a matching QP.

The Data Generation Module (DG) encompasses the end-to-end creation of synthetic exam papers, from sampling the SciQ and RiceChem datasets to calling the MSG and QPG after the data is generated.

We consider all of these elements together as they are not a part of our final programme and all work together with the singular goal of data generation

Each exam is stored as a sequential list of dictionaries, each one pertaining to a question. These question dictionaries can be classified into 3 types:

MCQ

```
"question_no": str, in-sequence question label,  
"q_id": str, sampled row id in SciQ dataset,  
"type": int, 0,  
"question": list, [0] is "question" column in sampled row, [1:5] are answer options  
made by shuffling "correct answer" and "distractors"  
"student_answer": str, uppercase alphabetical (A-D) representation of an option (A  
points to pos 0 etc). 50% chance to point to "correct answer", 50% chance to point to  
"distractor"  
"mark_assigned": 1 if pointing to "correct answer", 0 otherwise,  
"mark_scheme": ["A", "B", "C", "D"][pos "correct answer"],  
"max_mark": int, 1
```

Figure 32: MCQ Dictionary struct.

SF

```
"question_no": str, in-sequence question label,  
"q_id": str, randomly sampled row id in SciQ dataset,  
"type": int, 1,  
"question": list, [0] is "question" column in sampled row, [1:5] empty  
"student_answer": str, "correct answer" in row sampled 50% of the time, a "distractor"  
sampled the other 50%  
"mark_assigned": int, 1 if "correct answer" sampled, 0 otherwise  
"Mark_scheme": str, "correct answer" in row,  
"max_mark": int, 1
```

Figure 33: SF Dictionary struct.

LF

```
"question_no": str, in-sequence question label,  
"q_id": str, "[int of question, 1:5, of RiceChem question sample], s_id of answer  
randomly sampled"  
"type": int, 2,  
"question": list, [0] is the preprocessed question sampled from row 1, col 2 in Student  
Answers .csv for question  
"student_answer": str, preprocessed text answer from student "s_id"  
"Mark_assigned": int, Graded Rubric .csv for question sampled for s_id of answer,  
"mark_scheme": preprocessed mark scheme sampled from row 1, col 3 in Student  
Answers .csv for question  
"max_mark": Largest mar assigned in Graded Rubric .csv for question
```

Figure 34: LF Dictionary struct.

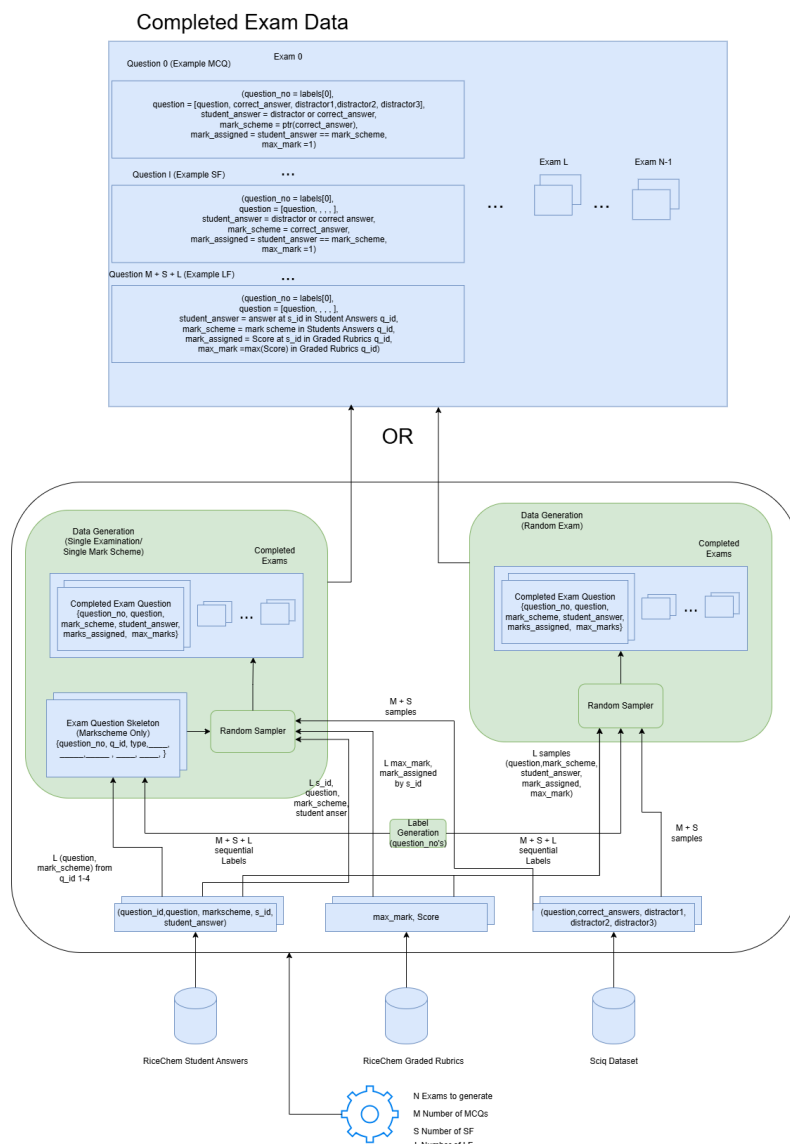


Figure 35: Flow diagram of how ground truth data for each exam is created from the datasets.

Data can either be generated randomly (Random Exam) with each question paper having its own mark scheme, or having a many QP-to-one-MS generation (Single Examination).

When Single Examination is implorded, a skeleton GT QP is created, in which data that would be consistent between uncompleted exams (Question Number Sequence, Questions, Maximum Marks Per Question) is generated once, but student generated data is kept blank.

Thereafter, to complete the symmetric data, our datasets (RiceChem, SciQ) would then be randomly sampled in accordance to the question content to create the Student Answer portion and hence a whole (or several) completed exam(s).

For each MCQ, we would keep the question and options the same, while making the student answer point to the SciQ “correct_answer” among the options or one of the distractors with an even occurrence of either during generation.

For each SF, we would keep the question the same while sampling that question’s “correct_answer” or a distractor with an even split between the two as well.

For each LF, we would keep the question (one of the 4 available in the RiceChem dataset) the same and simply randomly sample the dataset for a student answer for that question.

4.2.1 Pre-Processing

Though the RiceChem dataset is extremely rich, it does not fully encode the underlying student answers it is representing. The chemistry answers and even the mark schemes within the dataset include a lot of irregular characters and symbols, particularly in place of equations.

In these cases we have replaced them with “[indiscernible]” tag in order to make it clear during evaluation that this data is explicitly unusable (Bangalore and Joshi, 1999) and not a part of the answer.

Furthermore, when receiving the dataset the allocation of marks in the graded rubric was not explicitly specified. Rubric points, though being included in the Graded Rubric columns, are the only marking criteria, which in themselves are not consistent behaviour within or between questions. We adapt the mark schemes presented to the model from extrapolating the overarching behaviours in Section 4.2.2. We use the process of repeatedly singling out marks we are sure about to extrapolate weights and meanings of the remaining ones until we are sure of the whole rubric.

Lastly, we ensure all equations are “flattened” (see Section 3.3.3), where powers and subscripts were turned into plaintext representations. **Du et**

al. (2024) shows that LLMs have been shown to have robust capabilities in picking up more complex equation contexts from plaintext representations. Though this is not overtly useful with RiceChem and SciQ (due to its simplicity), it is essential if we ever want to use other more technical datasets.

4.2.2 Mark Scheme Interpretation

In order for the model to mark the LFs we had to adapt the mark schemes in a manner that could be readily understood by the model (Fan et al., 2025), significant changes had to be made to how the mark scheme was presented while ensuring we are not misrepresenting how the papers were marked in the ground truth data.

At first it was assumed that each line in the Mark Scheme section in the Student Answers section was worth exactly one mark. The Graded Rubrics were extremely irregular in how they awarded marks, with some behaving in the previously mentioned manner (see Section 6.2.1) while others had more dynamic marking behaviours.

Initial results (Fig 87) did not account for these behaviours, which include:

- Certain cases, like the column “Correct response” being checked, awarded full marks but meant all other marks were checked as false. Other cases had full marks being awarded but the inverse boxes being marked true (“Correct response” false but every mark being correct)
- Certain marks called the student to “PARTIALLY explain” or “FULLY explain” a concept. These marks were mutually exclusive, with them being TRUE-FALSE being worth one mark, and FALSE-TRUE being worth 2. This created a pitfall as a question that does not “partially explain” required concepts can still get full marks.
- Ambiguous Negative Marks, with one being awarded for an “Incorrect Statement Included” in Question 2 and “Core Charge Calculation Error” awarding -0.5, with the dataset omitting all equation data. Columns that signify no difference in mark assignment

Final model iterations (Fig 88) now account for these behaviours wording these behaviours into natural language understandable by our LLM evaluator.

			Explaining sentence 1: energy levels of an electron in an atom are quantized	Explaining sentence 1: FULLY explains energy/frequency absorbed must equal the difference in energy levels in an electron	Explaining sentence 1: PARTIALLY explains energy/frequency absorbed must equal the difference in energy levels in an electron	Explaining sentence 2: a minimum amount of energy is needed to eject an electron	Explaining sentence 2: any additional energy becomes kinetic energy	Incorrect statement included			
SD	Score	7	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
7986cfe-b65a-49b1-8219-85c											

Figure 36: Example of a Q2 Graded Rubric Row

Correctly states that frequency is proportional to energy of light
Explaining sentence 1: energy levels of an electron in an atom are quantized
Explaining sentence 1: FULLY explains energy/frequency absorbed must equal the difference in energy levels in an electron
Explaining sentence 1: PARTIALLY explains energy/frequency absorbed must equal the difference in energy levels in an electron
Explaining sentence 2: a minimum amount of energy is needed to eject an electron
Explaining sentence 2: any additional energy becomes kinetic energy

Figure 37: Example of Raw Mark Scheme Available in Student Answers Q2

1 mark: Correctly states that frequency is proportional to energy of light
1 mark: Explaining sentence 1: energy levels of an electron in an atom are quantized
2 marks: Explaining sentence 1: FULLY explains energy/frequency absorbed must equal the difference in energy levels in an electron
OR
1 mark: Explaining sentence 1: PARTIALLY explains energy/frequency absorbed must equal the difference in energy levels in an electron
1 mark: Explaining sentence 2: a minimum amount of energy is needed to eject an electron
2 marks: Explaining sentence 2: any additional energy becomes kinetic energy
1 mark: No incorrect statement included
Award 0 marks if answer blank

Figure 38: Adapted Mark Scheme for Student Answers Q2

4.3 Calligrapher

Line Vector Object

An object in motion stays in motion unless acted

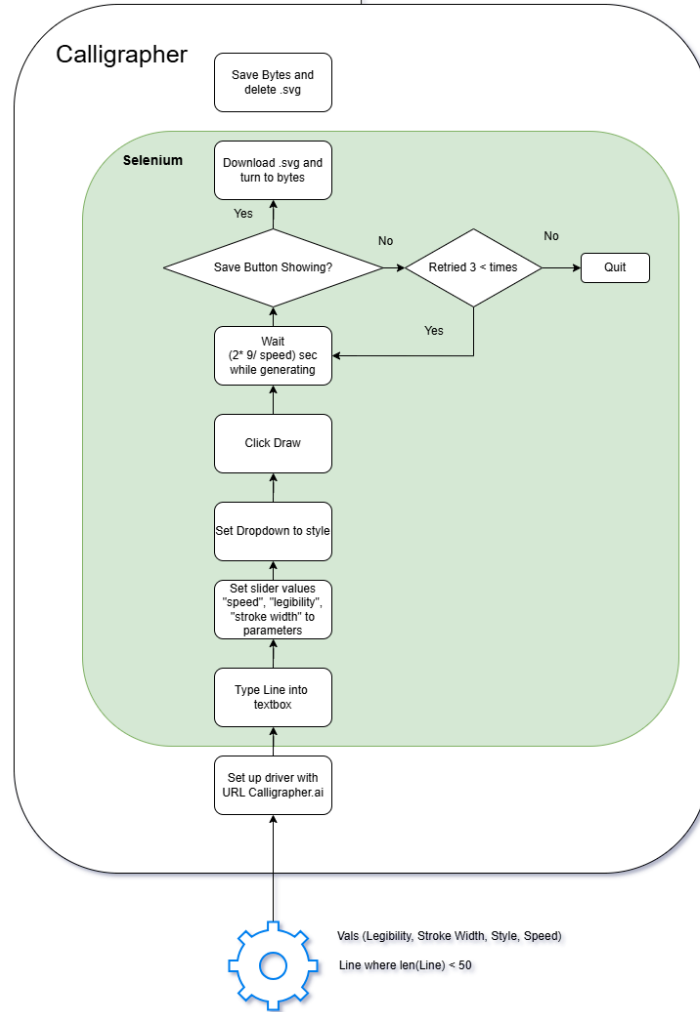


Figure 39: Flow diagram of the Calligrapher module

A vast majority of handwriting synthesis-only models were originally developed approximately 7-9 years ago (Díaz et al., 2025). Publicly available implementations typically using TensorFlow 1.x and operating within the now deprecated Python virtual environment versions (2.7, 3.3–3.6). These implementations (Carter et al., 2016; Daware, 2021), as well as the peripheral packages they depend on are no longer available through current package man-

agers such as pip, anaconda, or even third-party repositories.

Upon both experimentation with package variations as well as modifying the source code when implementing the Vasquez implementation of the Graves Model (2018), it became clear a more strategic approach in accessing the model outputs was required.

The creators of the project have a maintained demo platform called **Caligrapher.ai**, which allows real-time access to the model and control of the hyper-parameters. The website took a text (max. length 50), the parameters speed (1–9.51), legibility (0.15–2.5), stroke width (0.1–1.5), style (int 1–9).

To have programmatic access to desired results, we used Selenium to automate the inputting, setting of parameters, running and downloading of results once our textual input was submitted. As mentioned previously, Student Answers were wrapped to the letter of the last word before the 50 char limit. After submitting the input, our program waits $2 \times (9 \times \text{hyper_param}(\text{speed}))$ seconds while the model “writes” (Section 3.3.3) the output in real time, downloads the SVG, and pre-processes it for use within the paper.

Interestingly, the “legibility” parameter dial in this implementation controls the bivariate Gaussian component sampling bias (Graves, 2013), with 1 having extremely low variance and favouring more likely predictions, while a value of 0 has the model sampling at a higher variance. As the most probable sample would be the one that was seen in the handwriting in its training data, increasing this bias leads the model to favour more “legible” handwriting.

Furthermore, the “speed” parameter on the demo website is actually an accumulated value control (k in Graves paper (2013)) that controls how quickly the network’s attention (window) moves along the textual input.

It was within the interest of quick data generation to not change the speed hyperparameter as it could remove that speed advantage gained when automating the QP generation. We tested to see if increasing the speed parameter significantly changed handwriting behaviour and found no tangible difference.

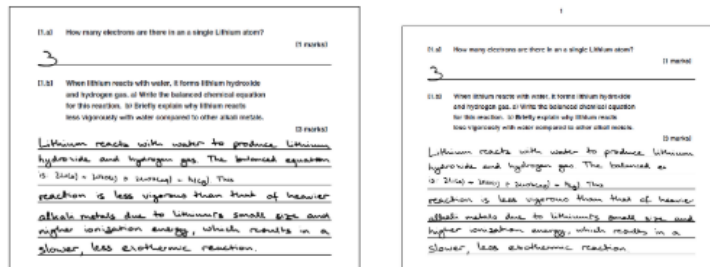


Figure 40: Speed param. does not seem to affect human readability enough to make it worthwhile to decrease generation speed

As stated previously, the model is known to have issues when outputting single characters (Section 3.3.3). A solution for this has been to append a string to the end of the letter string and to adjust the vector to only display up until a certain cutoff point.



Figure 41: A being input by itself



Figure 42: The letter "A" is legible when submitted in a larger input

Overall, the model was extremely robust in creating varied and human-like handwriting. Due to the programmatic hyperparameter control handwriting styles we also saw great success in the reproduction of handwriting for each separate line, creating the impression a single student was responsible for completing all answers.

It is hard to say if this approach is more efficient than running the model ourselves, but this part of the project is not user-facing and hence efficiency was not a concern.

4.4 Question Paper Generator

The role of the Question Paper generator was to take the symmetric GT data created, and to generate the PDF corresponding to the question sequence based on a series of layout rules.

The implementation of the QPG uses a greedy stateful procedural generation (Ferraioli et al., 2022) algorithm, where the input GT data is turned into a stack, and the page is generated through local decisions based on the current page layout, height and attributes of the top stack element. This approach allows for the QPG to start new pages when there is not enough space for a particular question, or when a long-form question overflows onto a following page.

There are 3 types of elements within the stack:

4.4.1 Textual question (0)

These are questions which have the student “writing a line” and the header for which has not yet been placed on the page. In the QPG, SF and LF questions can be handled the same as they only differ in data source and written answer length.

When printing a whole question (Question Number, Question, Max Mark, Student Answer), two parts have unknown length before printing due to line overflow: the Question and the Student Answer.

This means that before placing the question, the amount of page height left in relation to the amount of textual and handwritten lines is checked.

If there is not enough space for the question header and 2 lines of handwritten text, the whole textual question (type 0) will be pushed whole back onto the stack and a new page started.

If there is enough space for the header and two lines, the header will be placed onto the page, Calligrapher will be called to write the 2 lines, with the rest of the text being packed into a type 2 element (Text Block), pushed onto the stack and a new page started.

4.4.2 Multiple Choice Question (1)

Multiple choice questions cannot overflow onto the following page and hence will always start a new page if there is not enough space.

Unlike type 0 elements, they do not have variable length Student Answers but can still have unpredictable page usage due to overflow in the printed length of Question and Answer Options fields.

Calligrapher is called to write the letter corresponding to the Student Answer for that question.

4.4.3 Text Block (2)

These elements are only created when a type 0 element has enough space to place the header and 2 lines but overflows due to having leftover text. Type 2 stack elements contain no Question, Question Number or Marks header data.

When the GT Student Answer text is being wrapped before printing, new paragraphs (or double new lines) are added as an “`\n`” marker, which if fed into Calligrapher will simply create a line with no handwritten text on it.

If this textblock overflows again during its generation, a new page will be started and a textblock with the remaining data will be pushed onto the stack.

The ability to in theory create an “infinite text block” might not be realistic to the exam board design practices we are basing this off, but would still allow us to effectively test parsing in the case our dataset provides such answers (see **Section 3.4.2**).

Overall, this system is a powerful and flexible solution to create exam papers based on any ground truth data, with realistic handwriting behaviour and, due to its modularity, can be easily extended by simply adding new types of questions to the generation algorithm.

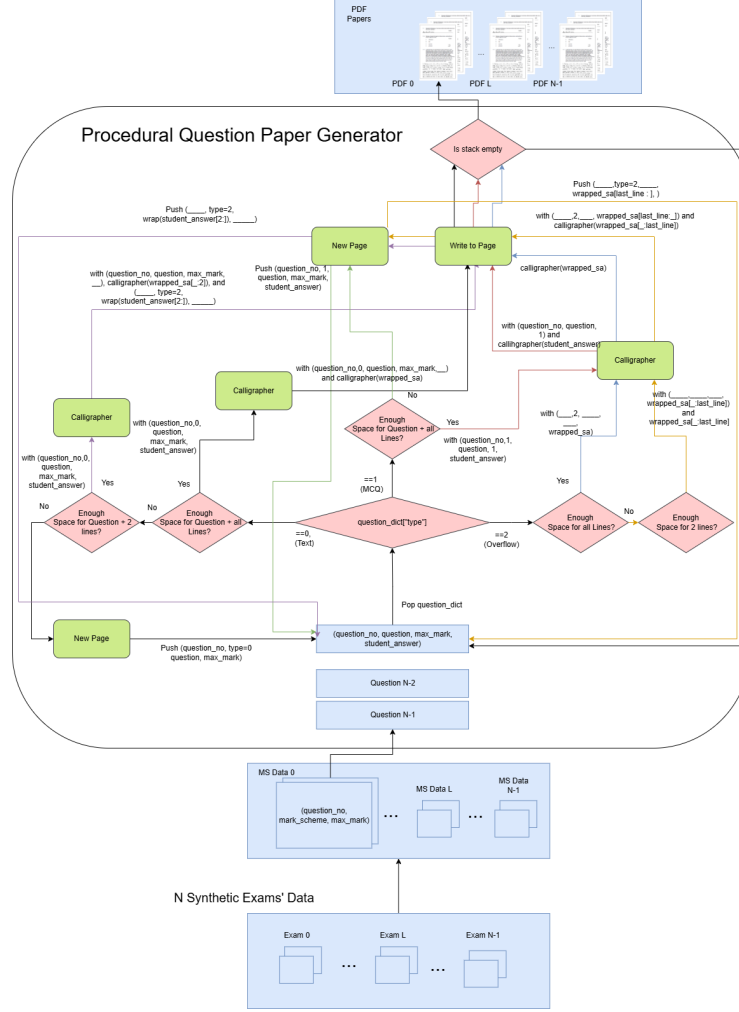


Figure 43: Flow diagram for the QPG

4.5 Mark Scheme Generator

The implementation of the MSG was a lot simpler as there was no handwritten element and the only variable part of any question was the Mark Scheme section ("Answer" section in OCR exam board).

Question Number	Mark Scheme	Max Mark
1.a	B	1
1.b	sapwood	1
2.a.i	- Correctly states that frequency is proportional to energy of light - Explaining sentence 1: energy levels of an electron in an atom are quantized - Explaining sentence 1: FULLY explains energy/frequency absorbed must equal the difference in energy levels in an electron - Explaining sentence 1: PARTIALLY explains energy/frequency absorbed must equal the difference in energy levels in an electron - Explaining sentence 2: a minimum amount of energy is needed to eject an electron - Explaining sentence 2: any additional energy becomes kinetic energy	6
2.a.ii	a basin	1
2.b	D	1
3.a	thicker fur	1
3.b.i	B	1
3.b.ii	- Sentence 1 is correct. Valence bond theory describes that atomic orbitals must be half-filled to participate in covalent bonding. - Sentence 2: Correct number of hybrid orbitals. In this molecule, carbon must form three hybrid orbitals to form three electron domains. - Sentence 2: Correct type of	7

Figure 44: An example for a mark scheme generated from question_no, mark_scheme and max_mark sections in the symmetric GT

To create each mark scheme page, the total height of available lines was calculated and each column was checked to see which would hit the page limit first, with overlap being moved to the next page without the printed lines.

This was done all at once as table creation is an in-built feature of the ReportLab library.

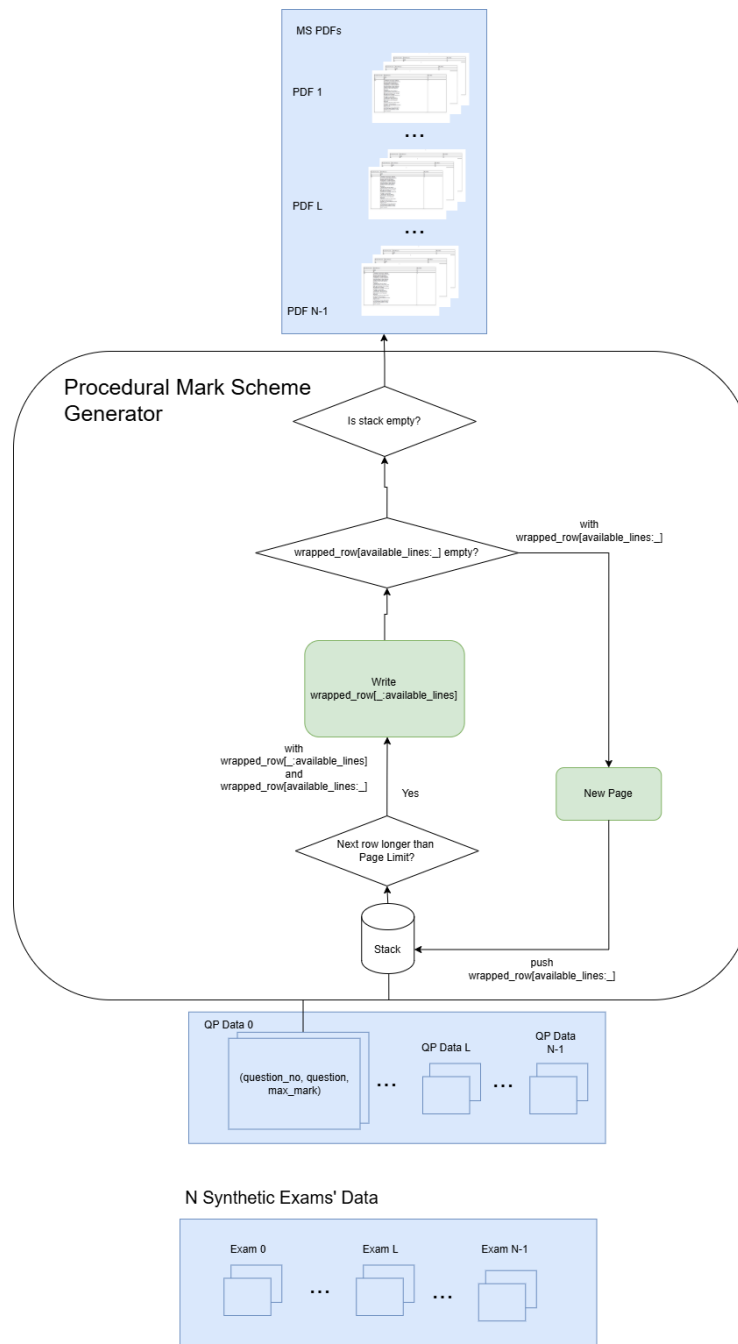


Figure 45: Flow diagram for the MSG

5 Development - Page Parsing

5.1 Page Parsing

5.1.1 Alternative Approaches

At the outset of the project it was planned to either create a bespoke parsing model or train an existing one such as DocLayNet (Dolfi et al., 2022).

Before the DG module was created, experimentation was done to acquire insights on possible approaches on the AQA papers. These papers, though downloaded directly from official sources, converted to images at an extremely high DPI, and appropriately pre-processed, were extremely hard to read, even by eye.



Figure 46: Extremely low resolution printing in the original copies of the AQA exams

Like in generation, the significantly more standardised and handwriting-free nature of mark schemes made them extremely easy to segment and conduct OCR on.

MARK SCHEME – GCSE BIOLOGY – 8462/04 – JUNE 2022

Question 1



Question	Answers	Extra information	Mark	AO / Spec. Ref.
01.1	nucleus	must be in this order allow chromosomes allow plasma	1	AO1 4.1.1.1 4.1.1.2
	(site of aerobic) respiration	allow makes ATP or releases energy do not accept produces / makes / creates energy do not accept anaerobic respiration	1	
	(cell) membrane		1	

Question	Answers	Extra information	Mark	AO / Spec. Ref.
01.2	photosynthesis	allow produce glucose / sugar allow to absorb (sun) light ignore contains chlorophyll	1	AO1 4.1.1.2 4.4.1.1

Figure 47: Extraction of tables from the AQA mark scheme, where each question was its own table, after which each table was converted into a csv for subsequent preprocessing

Question	Answers	Extra information	Mark	AO/
		allow produce glucose / sugar		AO1
		allow to absorb (sun) light		4.1.1.2
1.2	photosynthesis	ignore contains chlorophyll		4.4.1.1

Figure 48: The table extracted from Fig 47 using an early parsing algorithm. Note the lack of value in the mark column when it was present in the original segment

Experimentation with AQA Question Paper was minimal, as we did not yet have the training data and was moved to after the development of the DG module.

5.1.2 Question Paper Parser

To parse each Question Paper PDF each paper was converted into a series of images at 300 dpi vector scan.

Each image was then batched into pairs and passed in with the following prompt:

“

You are looking at 1 or 2 pages from an exam paper. These pages are sequential. In these exams there are both multiple choice and written questions. For each question, there are 4 elements in parsing order:

- 1. Question Number: usually in square brackets such as "[1.a.ii]" or "[3.1]".*
- 2. Question: the printed text of the question.*
- 3. Maximum Marks: usually in square brackets, e.g. "[1 marks]" or "[8 marks]".*
- 4. Student Answer: the handwritten response; if it is a multiple choice question, a single letter such as "A" or "D".*

-Return a JSON array where each object has headers question_no, question, max_mark, and student_answer.

-Treat each page chunk separately.

-If you see handwritten lines at the top of a page with no Question Number, Question, or Maximum Marks (only Student Answer lines), This indicates that a textual question is overflowing onto another page. Instead of creating a separate JSON object,

-Fuse these overflow lines into the student_answer of the previous question object by appending them with a newline.

-Do not cut any textual answer short.

“

Figure 49: QPP prompt

We instruct the model to format outputs as JSONs not only because we are already using Python dictionaries as our primary data medium for both GT and output sequences, but also because the model is extremely unlikely to break the overarching JSON schema when creating outputs (**Wedholm, 2024**), i.e., by leaving brackets open.

We remove padding around the JSON struct in the output text and use the Python JSON library to parse our data, in which each question is its own object. Text blocks overflowing onto the next page are concatenated onto the text of the previous page and all of the data is returned.

When using the GPT-4o’s multi-modal capabilities (**Shahriar et al., 2024**) in the Question Paper Parser (QPP), we encountered some issues when submitting whole exams into a single prompt, in which the model would, very often in crucial areas, shorten answer data while maintaining JSON integrity.

```
[
  ...
  {
    "question_no": "[2.]",
    "question": "What is the time complexity of binary search?",
    "max_mark": "[2 marks]",
    "student_answer": "..."
  }
  ...
]
```

Figure 50: Example omission of crucial data such as the student answers in order to stay within output limit

In theory, 4o-mini boasts a context window of 128k tokens, with each having an output limit of 16,384 (**PromptHub, 2024a**), and the ability to take a maximum of 10 images as input.

An A4 page at 300dpi (the resolution we use) would result in 8,699,840 vectors per image, in which patch embedding (**Mao et al., 2022**) is used to reduce 16x16 non-overlapping segments of the image into vector space. This reduces such an image to 33,945 tokens which allows for roughly 3 images to be reliably processed at once.

None of the exams we submitted were 10 pages or more in length and we were still encountering the aforementioned issues, in which after around 3 pages the model would begin to be “lazy” (**Liu et al., 2025**) with its output and start finding ways to not output the full parsed data.

In order to not overextend the model’s output and processing capabilities, we have chosen to submit 2 images for parsing at a time.

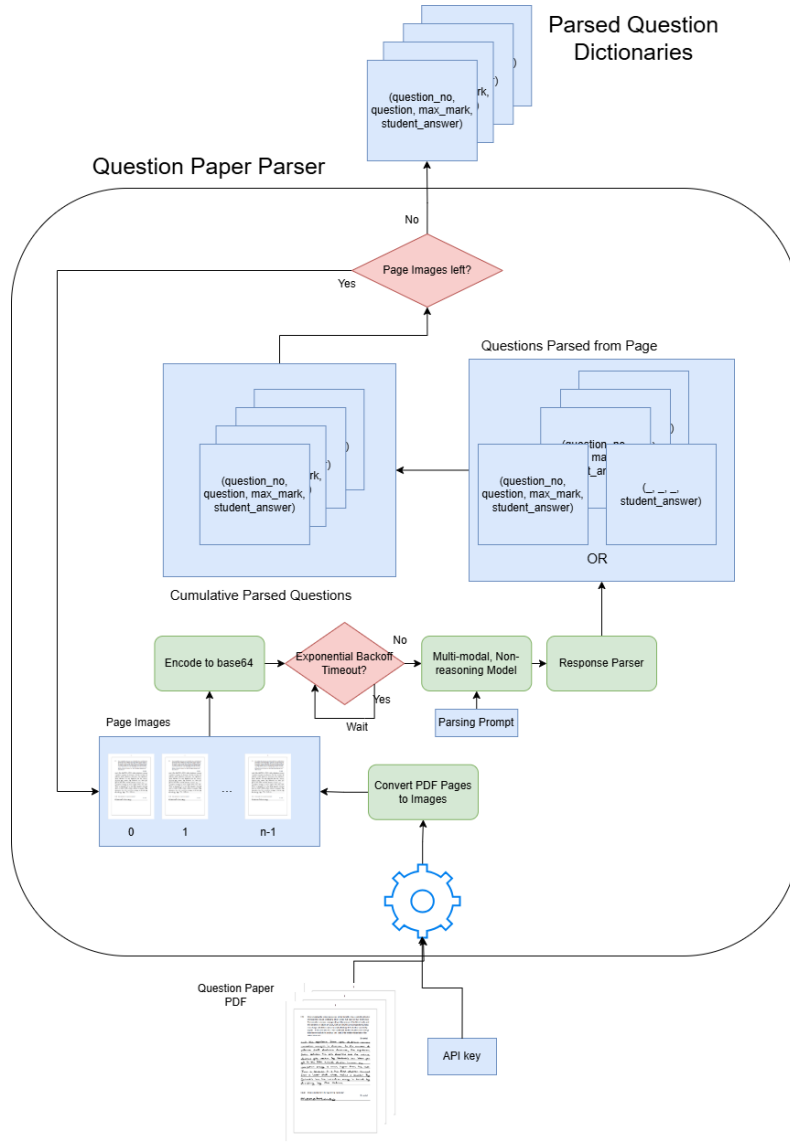


Figure 51: Flow diagram for the QPP

5.1.3 Mark Scheme Parser

When parsing the mark scheme, the same principles of table segmentation and Tesseract text to string mentioned in Section 5.1 were applied.

5.2 Parsing Effectiveness

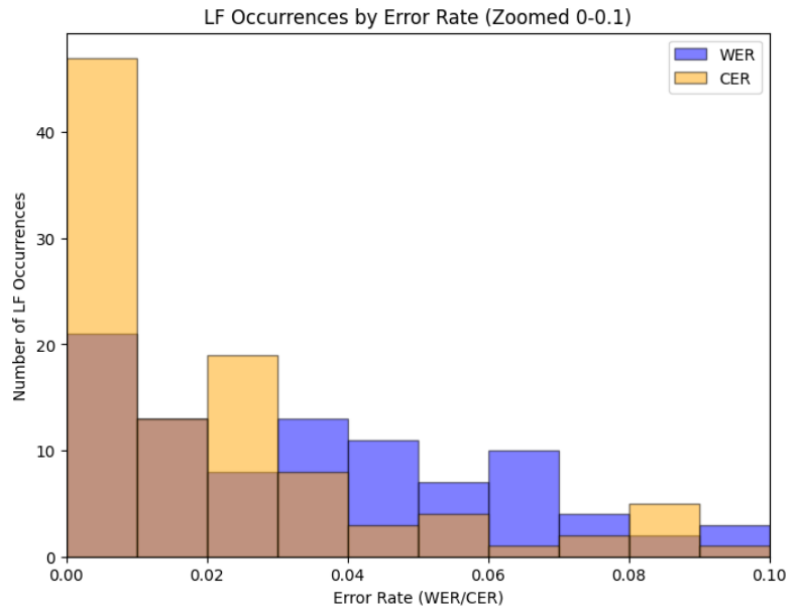


Figure 52: CER and WER for Long-Form answer parsing

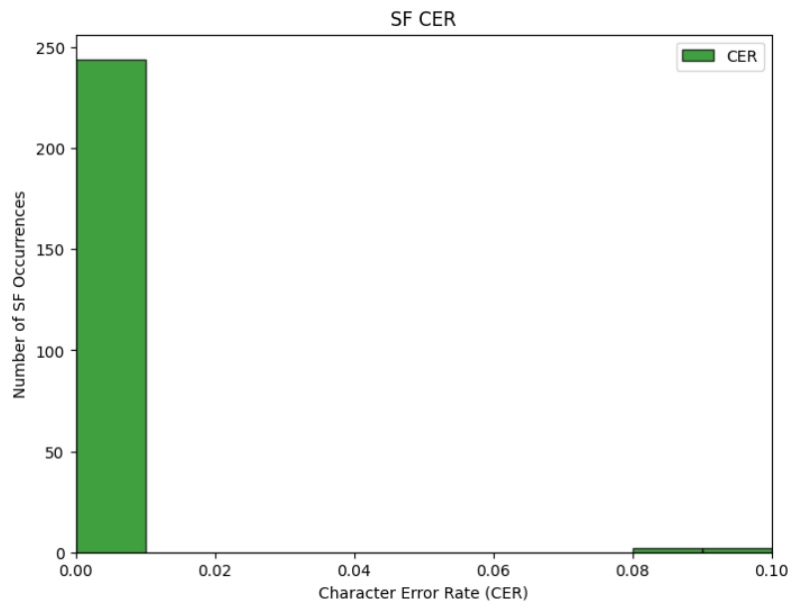


Figure 53: CER for Short-form answer parsing

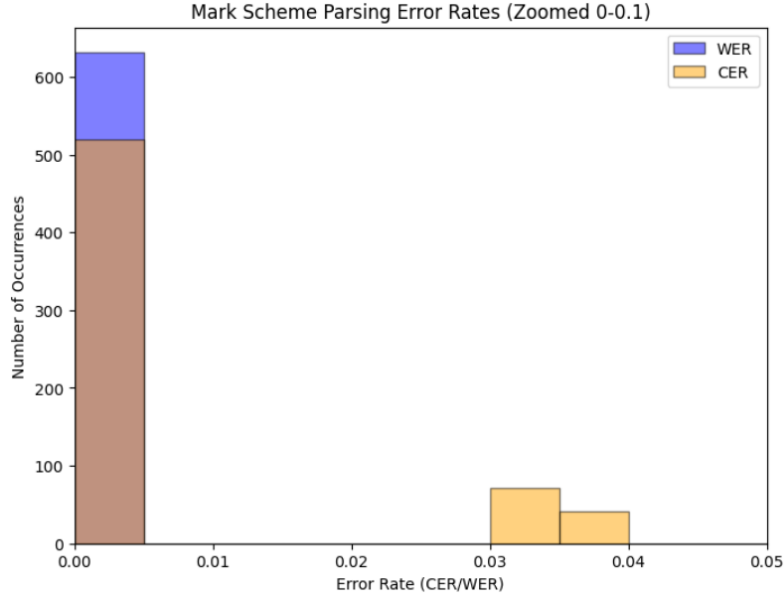


Figure 54: CER and WER for Mark Schemes

The error rates were calculated using CER and WER metrics. These are found by finding the Levenshtein distance (**Liu et al., 2006**), the number of insertions, deletions and substitutions that need to transform one character/word sequence into another, between the GT and the parsed output and normalised to output length.

It can be seen that the 4o model was able to parse the handwritten texts with sufficient results. For comparison, state of the art bespoke DNN OCR models such as (**Bluche et al., 2017**) performs at a mean CER of 9.96% and a WER of 15.80%, with our model achieving a rate of 11.71% CER and 14.16%.

Both the SF parsing and the mark scheme parsing were effectively 100%. The CER in SF was almost 0, likely due to the fact that the LLM was able to correct spelling in cases when the context of the question was understood from data parsed in the same question.

Despite the target texts being extremely short, MCQ parsing accuracy was relatively low at 82%. The size of the Student Answer and high chance of unforeseen illegibility issues with single-letter inputs from the Synthesis module (**Vasquez, 2018**), some answers letters were simply parsed as another letter entirely. In many cases, and likely due to the comparative size difference

between the answer box and other answer sheet elements, other parts of the question were deemed to be the “Student Answer” by the model, such as the entire Answer Options list.



Figure 55: Bar chart of MCP parsing success, with anything but the GT Student Answer within the parsed output being counted as “Incorrect”

6 Development - Evaluator Module

6.1 8.1 Reasoning Models and Chain of Thought

With our project we wanted to test out the capabilities of new Reasoning Models, which are able to extract key decision-making information out of textual prompts, and conduct numerical, logical or contextual analysis (**OpenAI, 2024**). Reasoning models go beyond traditional NLP/semantic analysis tools by incorporating reasoning mechanisms into the model’s core weights (Besta et al., 2025) training on the correctness of the reasoning process itself.

“**RLMs**”, as they are called in Besta et al. (**2025**), use the user input as the root of a reasoning structure, with reasoning steps simultaneously exploring and generating the tree, with the conclusion of the reasoning process being reached at one of the leafs.

In the case of OpenAI's o1 and o3 series, the model would even in some cases encode problems as runnable code and executing where possible in order to minimise hallucinations in the final output (**OpenAI, 2024**).

Li et al (**2023**) has also shown that these models have been shown to, in some cases, have PhD level performance in STEM fields such as Biology, Chemistry, Physics and others, which is unequivocally useful in our problem space.

6.2 Evaluation Algorithm

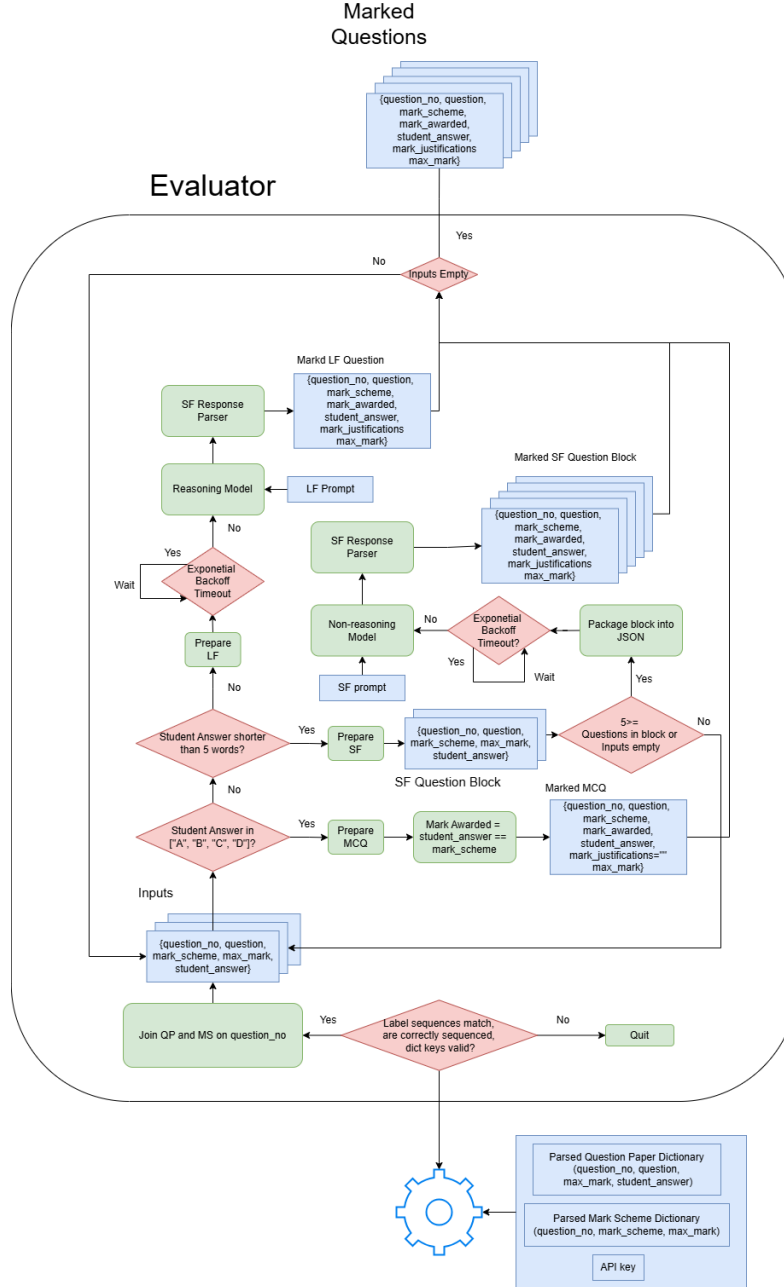


Figure 56: Evaluator module flow diagram

The evaluator requires two lists of question dictionary objects resultant from QP and MS parsing as input. They are assumed to be correctly structured, but not sequenced.

Each Question has Question Numbers stripped of all space and period (.) padding, which are then used as a key to join the QP and MS parsing output dictionaries. Each question is then split based on its type (SF, MCQ, LF). At the outset of the project it was proposed to use the reasoning model for every type of question but was deemed computationally wasteful (**See Section 6.4**).

MCQs are assumed to be single letter answers and are marked without the use of any external models, simply using string matching to the letter A,B,C,D, with all other data being ignored.

As SFs are usually one word, and never longer than 5, we use the non-reasoning model for evaluation. Performance is less likely to suffer (if at all) from having the model downgraded for a more compact one due to the lack of practical knowledge application within answers. This design choice is supported by findings from Khalid et al. (**2024**), who show that non-reasoning LLMs can perform well on tasks with low reasoning complexity requirements, often outperforming more recent reasoning models on straightforward relation prediction.

The reasoning model will be used primarily within the LFs due to the highly dynamic nature of the mark scheme and average answer length. Our dataset would be testing the model’s ability to simultaneously look for the presence or negation of a number of marking criteria with varying levels of abstraction.

SF and LF completions are made to provide a list `mark_justifications` of the same length as the amount of marks awarded, in which each mark points to a part of the mark scheme, using a Chain of Thought prompting which as been proven to prevent LLM hallucinations (**Wei et al., 2022**) and will provide a second layer of assurance of that the model is drawing from the mark scheme.

Each API call is made using exponential backoff in order to not fail on rate limit errors. As we are not running parallel processes we use the deterministic version of the Tenacity backoff function which, upon an encounter with a rate limit error, will wait an $2^x \cdot 1$ sec where x is the number of retries.

6.3 8.3 Prompt Engineering

Non-reasoning prompt:

```
You are an exam marker and must mark a series of short questions\nInputs are in JSON format...\n question_no: leave as is\n      question: the question the student is answering\n      student_answer: the student answer to the question\n      mark_scheme: the correct answer and marking guidance for the\nexaminer\n      max_mark: the maximum amount of marks the student may be\nawarded for the question\nYour job is out OUTPUT A JSON FORMAT RESPONSE for EACH\nQUESTION that can be parsed:\n      \"mark_awarded\": <integer: The final assigned mark>,\n      \"mark_justifications\": [\n      \"<string: Justification for mark 1>\",\n      \"<string: Justification for mark 2>\",\n      ...]\n\nExclude any triple quotes or \"json\" string at start. Make sure to not\njustify more marks than you've awarded.\nAdditional comments are helpful for when you want to justify not giving\nmarks\n\nInput:
```

Figure 57: Evaluator module flow diagram

Reasoning Prompt:

You are a Year 10 Teacher/Examiner. Your job is to provide a fair and justified judgement on a textual representation of a student answer based on criteria specified in the mark scheme provided with the question.

You will be given 5 pieces of information

```
question: "<string: the question the student is answering>"
student_answer: "<string: The student's answer>",
mark_scheme: "<string: The mark scheme>",
additional_guidance: "<string: Additional edge-case handling guidance>",
max_mark: <integer: The maximum number of marks available>
```

To be parsed, your output must have 3 fields with the exact format of:

mark assigned: "<integer:The final assigned mark>"

marks: ["<explanation for mark 1>", "<explanation for mark 2>", ..., "<explanation for mark n>"]

additional comment: "<Additional comments or an empty string>"

General Guidance:

- Do not allow students to gain the marks for a single mark scheme bullet-point several times

Output Mark justifications

- Each mark awarded has to be justified by you in the form of a string entry in the "marks" list

- Do not award marks for tasks not present in the markscheme

- Justifications for each mark should not be full sentences, only short facts like "referenced Moore's Law" or "Provided final answer of 67"

Figure 58: Evaluator module flow diagram

6.4 Cost/Efficiency

As this part of the project is user-facing, we have ensured to optimise the software for compute (monetary) and time costs.

As mentioned before, MCQs are handled locally and do not make any API calls due to their simplicity.

Furthermore, due to the short question lengths (from SciQ), 1–5 word answers, and single-word mark schemes, SFs were batched into completions in multiples of 5 by concatenating them into a single input JSON. We did not increase it past this because the output format was one of JSON with each marking result being a contained object, which despite content brevity had the potential to needlessly stretch the context window and therefore the model's analytical capabilities (Mervaala, Kousa 2025).

Like 4o-mini, o1-mini has a context window of 128k tokens, a 65,536 token output limit (PromptHub, 2024b). Though our longform answers fall well

inside this limit, it was a deliberate choice to submit one at a time in order to not over-extend the model’s attention (Qin et al 2024), which degrades parts of the text well before reaching the model’s context limit.

The end-to-end marking of an exam paper with 4 MCQs, 4 SFs, and 2 LFs cost £0.027, with the majority of the costs being incurred by the use of the 4o model, which handles not only the marking of SFs but also the QP parsing. The overall ratio of the costs is around 1.39:1 between the 4o and o1 models respectively.

It took 8:51 min to mark 15 papers which equates to about 35.4 seconds per unit.

With the assumption that each MCQ and SF are worth 1 mark and each LF is worth 8, it means a 100-mark exam would cost us around 5 times that: £0.135, with a class of 30 costing £4.05.

Using the modest average assignment marking time of 9.5 minutes we collected in the Charter School North Dulwich Survey (see **Section 2.1**) and figures from a report from the Department of Education (**2024**), which states the average annual salary for secondary school teachers in non-leadership positions outside of London is £31,650 (with London teachers averaging at £38,766), which we can calculate to be an hourly rate of £16.40.

This means under perfect conditions and using our extremely modest marking time estimate, a teacher may take 4 hours and 45 minutes, costing £77.90 for a single exam for a classroom of 30 students. This means our tool is at a minimum of 19.23 times more cost effective and 32.20 more time-efficient than a teacher marking these tests.

In future iterations of this tool it would be a priority to replace the SF marking software with a more compact, potentially even locally-run model, in order to cut this cost down further.

The parsing module could also be replaced but would certainly create large amounts of technical overhead with heavy adaptations needed for every domain or new exam layout. The decision to use this approach and its trade-offs would have to be evaluated by the implementing user.

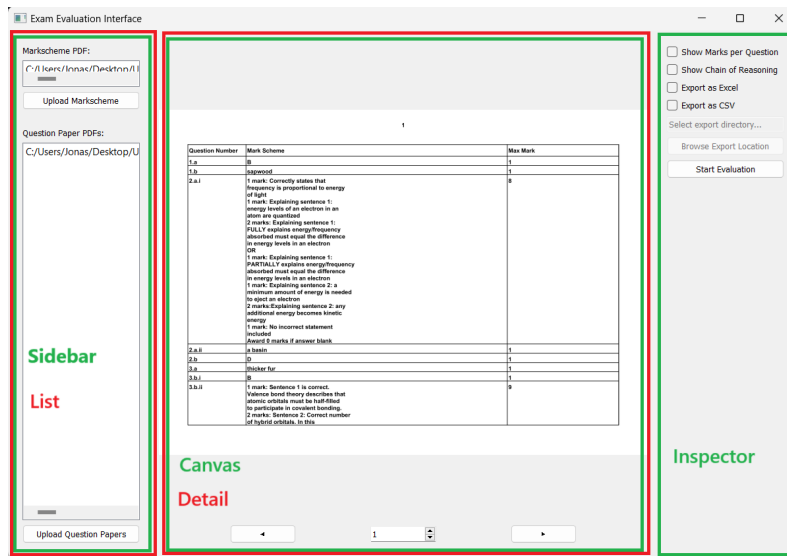


Figure 60: Design paradigm segmentation of GUI

This segmented structure gives a clear sequence of interactions to facilitate the user's end goal. The intended order of actions follows a clear right-down hierarchy, MS and QP upload being in the first pane, the canvas allowing the user to inspect all PDF pages post-upload, and the after set configurations before processing and receiving their outputs.

7.2 Requirements Fulfillment

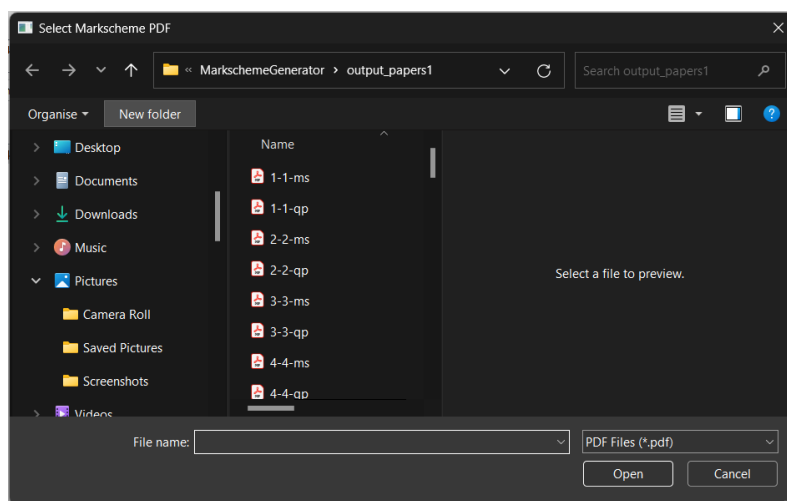


Figure 61: File Explorer which opens upon pressing "Upload Mark Scheme"

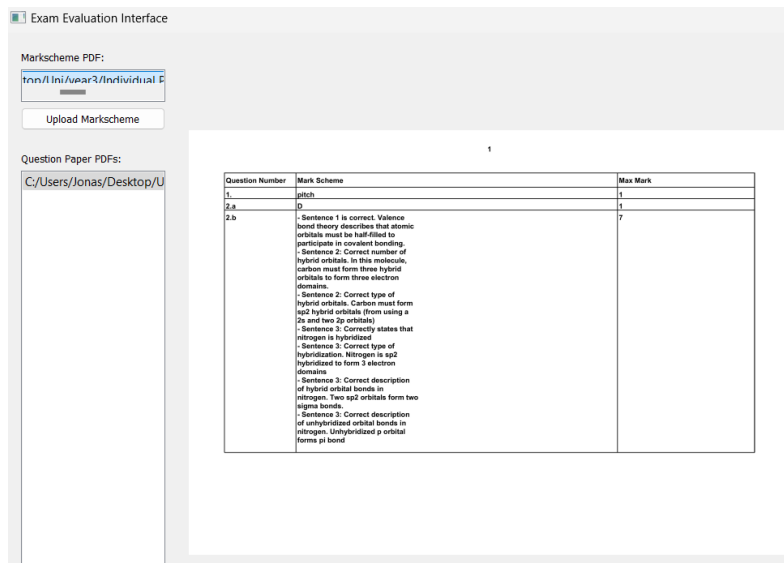


Figure 62: GUI after the mark scheme has been loaded

7.3 Requirement Fulfillment

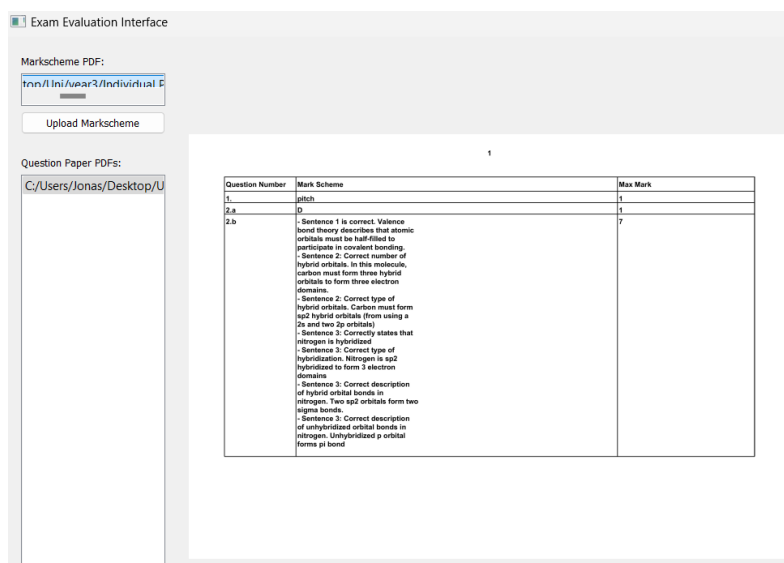


Figure 63: GUI after the mark scheme has been loaded

No invalid uploads allowed

Only directories and PDF files visible to select in explorer

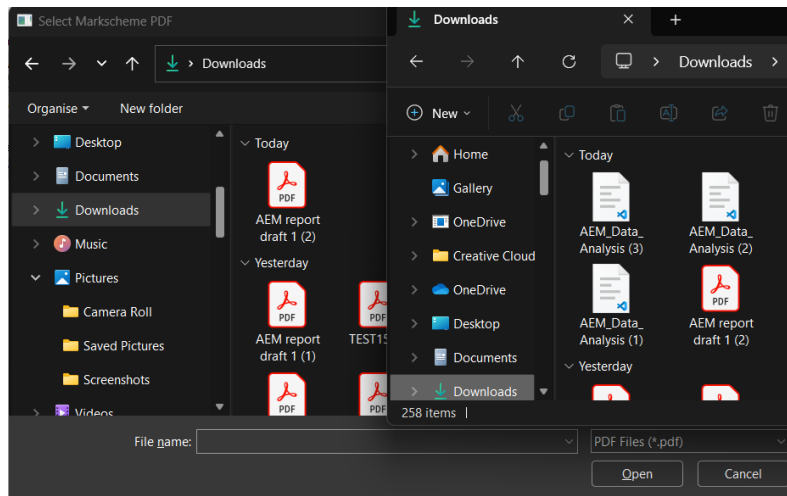


Figure 64: The same folder containing various file types being viewed from within and outside of the mark scheme selection file explorer

Upload in Answer Section Opens files in explorer for answers

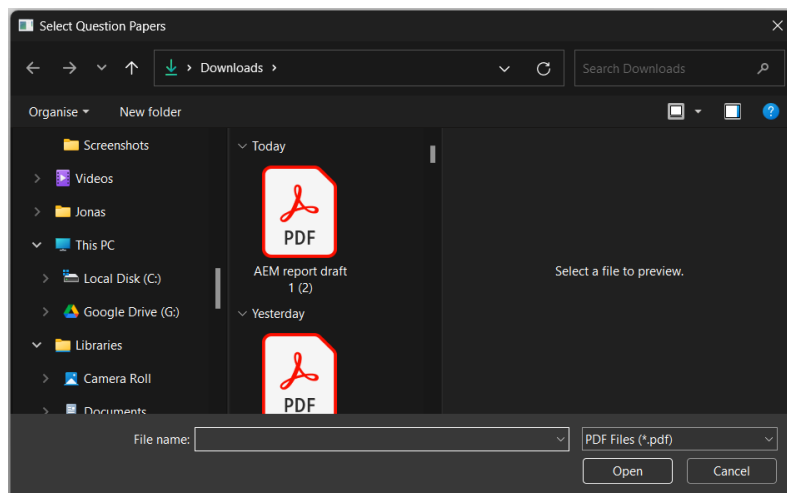


Figure 65: File Explorer which opens upon pressing “Upload Question Paper”

Successful PDF upload

Question Paper PDFs:

C:/Users/Jonas/Desktop/U

direction

[2.a] What group of single-celled organisms lacking a nucleus comprises the most numerous organisms on earth? [1 marks]

A. carbohydrates

B. fungi

C. eukaryotes

D. prokaryotes

[2.b] A CHEM 121 student was asked what hybrid orbitals must be present to form methanimine (CH₂NH), for which a correct Lewis structure is shown below. The student responded: According to valence bond theory, Carbon cannot form four bonds because it only has two unpaired valence electrons. So, it has to form four sp³ hybrid orbitals to create the four bonds. Nitrogen doesn't need to hybridize because it already has three unpaired 2p valence electrons to form the three bonds with Carbon and Hydrogen. Assess the accuracy and logic of the student's response: briefly explain whether the reasoning presented is logical, noting what information is correct or incorrect and providing correct logical reasoning and explanation where needed. This question can be reasonably answered in 150 words or fewer. [7 marks]

1. Pretty much true, valence bond theory claims that atoms only form bonds in orbitals that are half-filled, and carbon only has 2 half-filled orbitals. 2. It does not form sp³ orbitals

Figure 66: GUI after the question paper has been loaded in

Batch upload

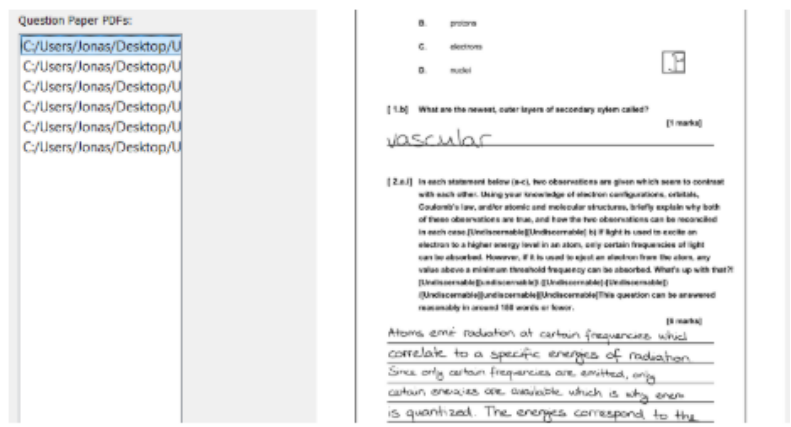
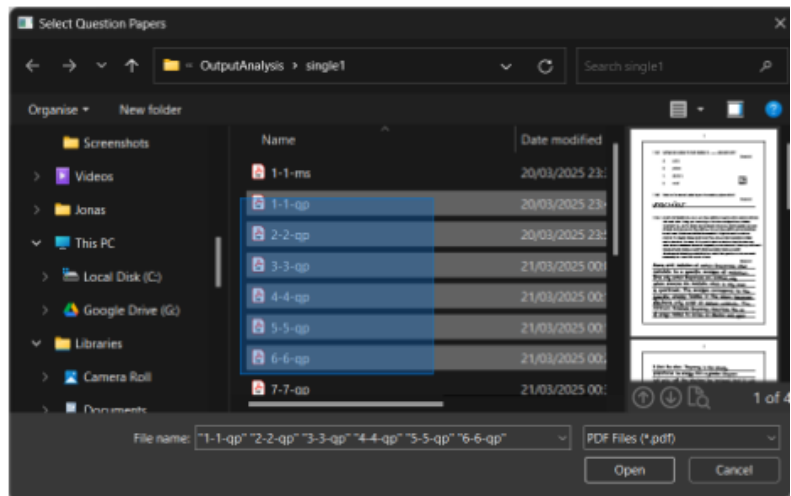


Figure 67: Multiple question papers being uploaded

No invalid uploads

Only directories and PDFs visible

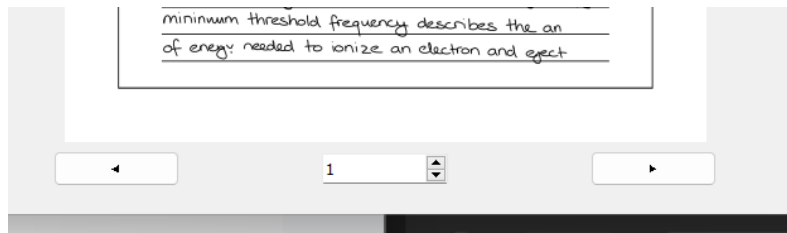


Figure 70: Inspector navigation arrows

Visualiser blank on open

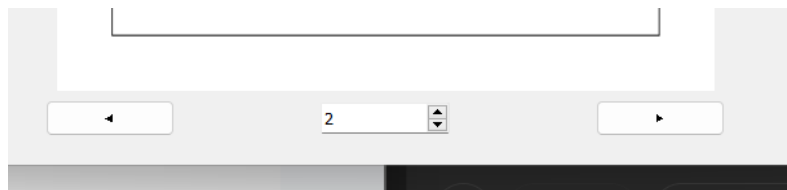


Figure 71: Inspector when no file has been uploaded

Page navigation work within page range

Numbers outside of page range cannot be typed, arrows do not work past limit. Visual demonstration not possible in static form.

No navigation outside of page range

As above.

Invalid start

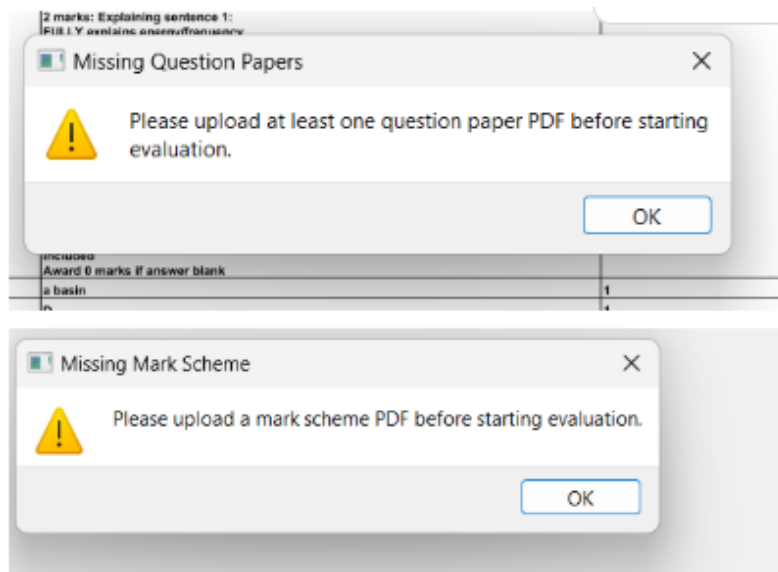


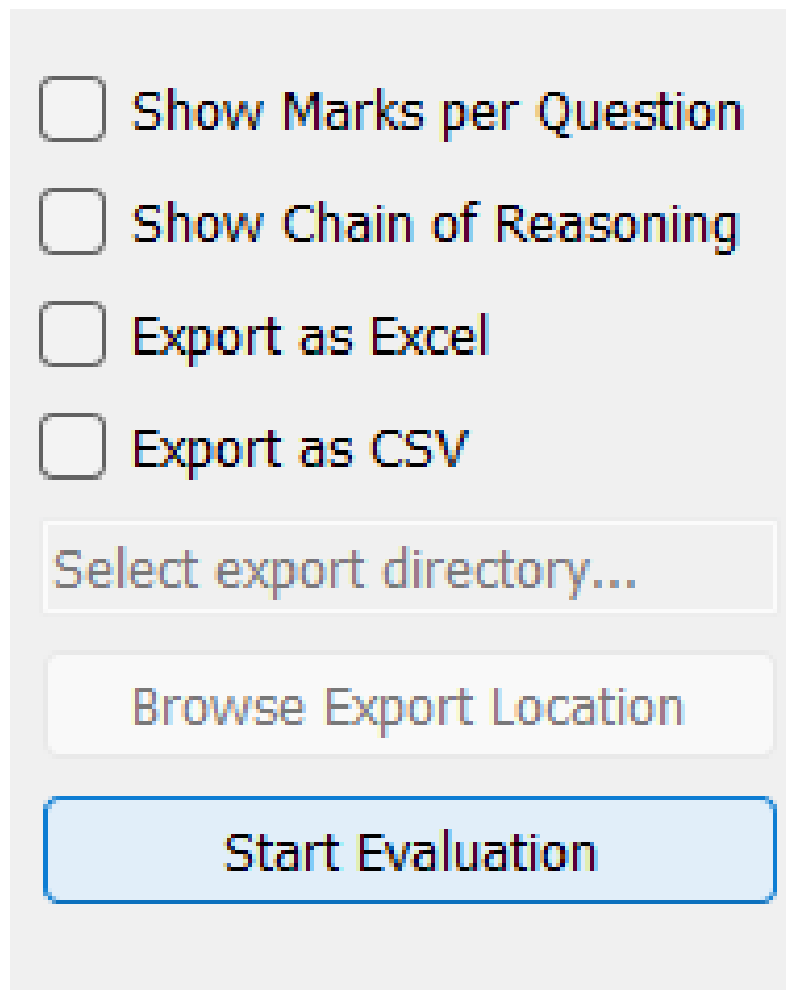
Figure 72: PDF upload errors

No configurations selected outputs only the final mark

	A	B	C	D	E
Question P	Marks Awarded				
1_1-1-qp.p		13			

Figure 73: Marking results of “question paper 1-1” with “mark scheme 1”

Cannot select Target directory without choosing to CSV/Excel export option first



The image shows a user interface with a light gray background. It contains four unchecked checkboxes stacked vertically, each followed by text: 'Show Marks per Question', 'Show Chain of Reasoning', 'Export as Excel', and 'Export as CSV'. Below these is a text input field with the placeholder text 'Select export directory...'. Underneath the input field is a button labeled 'Browse Export Location'. At the bottom is a large, rounded rectangular button with a blue border and light blue background, labeled 'Start Evaluation'.

☐ Show Marks per Question

☐ Show Chain of Reasoning

☐ Export as Excel

☐ Export as CSV

Select export directory...

Browse Export Location

Start Evaluation

Figure 74: User needs to select an export option to have access to file explorer

No scanning if “Export as CSV” selected if there is no Target Directory

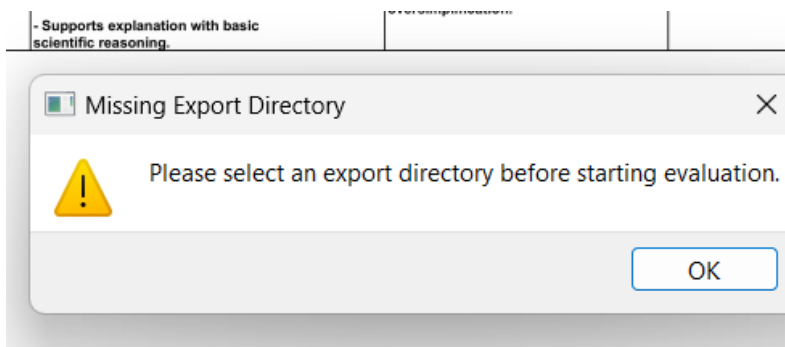


Figure 75: Error upon user selecting export mode and not selecting a destination

Can successfully export results as CSV and
 “Show Marks Per Question” scan runs successfully and
 “Chain of Reasoning” scan runs successfully

question_n	question	ident	ans	mark_sche	mark_ward	max_mark	justificational	comment
1.a	Isotopes a C	B		0	1			
1.b	What are t vascular	sapwood		0	1			["The stud The student misunderstood the terminology related to xylem layers.
2.a.i	In each stz Atoms emi - Correctly		4	6				["frequency proportional to energy of light", 'energy levels are quantized', 'partially explained energy/frequency
2.a.ii	What is a t a basin	a basin	1	1				["The student's answer 'a basin' matches the correct answer."]
2.b	Which part C	D		0	1			
3.a	What do s thicker fur thicker fur		1	1				["The student's answer 'thicker fur' is correct."]
3.b.i	The intent B	B		1	1			
3.b.ii	A CHEM 11: The first sc - Sentence		0	7				The response does not address the required criteria in the mark scheme.
4	The group C	C		1	1			
5	What does oil	gas		0	1			["The stud The student may have confused the state of matter with a substance.

Figure 76: Evaluation results export of a single paper with “Show Marks Per Question” and “Show Chain of Reasoning Enabled”

8 Experiment: Final Performance Evaluation of AEM Pipeline

The final benchmark consisted of a series of 5 distinct examinations with their own mark scheme. The first four examinations contained 15 completed tests each, with the last one consisting of 7. In total there were 67 total tests, with 268 of the questions being MCQs (Multiple Choice Questions), 268 being SFs (Short-form Questions) and 134 LFs (Long-form Questions). Each test was created over the course of 8 hours using randomly seeded Data Generation sequence on Single mode. Answers were sampled from the datasets as outlined in **Section 4.2**, with the Ground Truth data being saved alongside the generated PDFs.

For each exam the MS (Mark scheme) and QPs (Question Papers) were then input into the software and ran with “Show Marks Per Question”, “Show Chain of Reasoning” and “Export as CSV” enabled.

When analysing the data, each output CSV file was paired up with its Ground Truth csv data, with data being created through a pairwise join on the `question_no` column.

When running the experiment 2 out of 67 tests were lost.

Lost Exam 1:

The Question Paper generation algorithm checks for height limits as before and during generation of a question. It appears that it erroneously succeeded the first check, but failed the second, carrying the header onto the second page. This question was read twice and failed the question number check, which is crucial for successful marking. This error was due to a bug in the exam generation and provided no insights into the final software.

```
Error Evaluation: Length of Markscheme and Question Paper keys do not match
QP Keys['1.a', '1.b', '2.a.i', '2.a.ii', '2.a.iii', '2.b', '2.b', '3.a.i', '3.a.ii', '3.b.i', '3.b.ii']
MS Keys: ['1.a', '1.b', '2.a.i', '2.a.ii', '2.a.iii', '2.b', '3.a.i', '3.a.ii', '3.b.i', '3.b.ii']
```

Figure 77: Question Number matching error

[2.b] When studying the emission sources within the Milky Way, a satellite detected interplanetary clouds containing silicon atoms that have lost five electrons.b) The ionization energies corresponding to the removal of the third, fourth, and fifth electrons in silicon are 3231, 4356, and 16091 kJ/mol, respectively.Using core charge calculations and your understanding of Coulomb's Law, briefly explain 1) why the removal of each additional electron requires more energy than the removal of the previous one, and 2) the relative magnitude of the values observed.

[8 marks]

3

[2.b] When studying the emission sources within the Milky Way, a satellite detected interplanetary clouds containing silicon atoms that have lost five electrons.b) The ionization energies corresponding to the removal of the third, fourth, and fifth electrons in silicon are 3231, 4356, and 16091 kJ/mol, respectively.Using core charge calculations and your understanding of Coulomb's Law, briefly explain 1) why the removal of each additional electron requires more energy than the removal of the previous one, and 2) the relative magnitude of the values observed.

[8 marks]

Figure 78: Repeating question header placement due to Data Generation error

Lost Exam 2:

The second anomaly was considerably more worrying. The parsing model seemed to consistently pick up an extra question “2.a.ii” or “2.c.i” with absolutely no obvious visual cues. This could possibly be due to some artifact created during image encoding and is a testament to the underlying unpredictability of LLM integration into automation pipelines. Given an extended timeline for the project, investigation into the root cause of this and error handling for similar cases would be necessary.

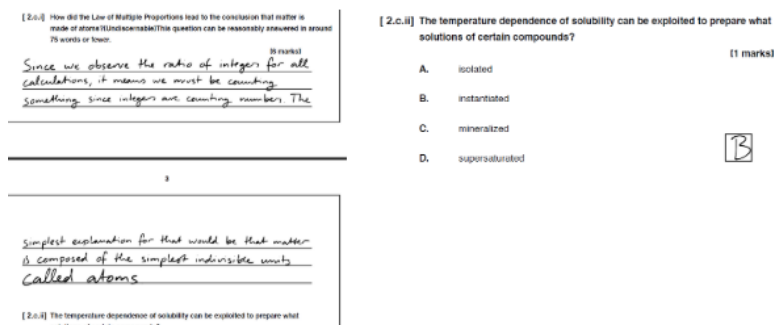


Figure 79: The two questions in between which the phantom question was consistently parsed

```
Error Evaluation: Length of Markscheme and Question Paper keys do not match
QP Keys['1.a', '1.b.i', '1.b.ii', '1.c', '2.a', '2.b.i', '2.b.ii', '2.b.iii', '2.c.i', '2.a.ii', '2.c.ii']
MS Keys: ['1.a', '1.b.i', '1.b.ii', '1.c', '2.a', '2.b.i', '2.b.ii', '2.b.iii', '2.c.i', '2.c.ii']

QP Keys['1.a', '1.b.i', '1.b.ii', '1.c', '2.a', '2.b.i', '2.b.ii', '2.b.iii', '2.c.i', '2.c.ii', '2.c.iii']
MS Keys: ['1.a', '1.b.i', '1.b.ii', '1.c', '2.a', '2.b.i', '2.b.ii', '2.b.iii', '2.c.i', '2.c.ii']
```

Figure 80: Resultant question number matching errors

After pairing, the data points resulting from pairing input and output data is then siloed into LFs, SFs and MCQs for further processing.

Due to both SFs and MCQs being single mark questions we were treated as Classification problems, with the Predicted value being the evaluators result and the true mark being whether

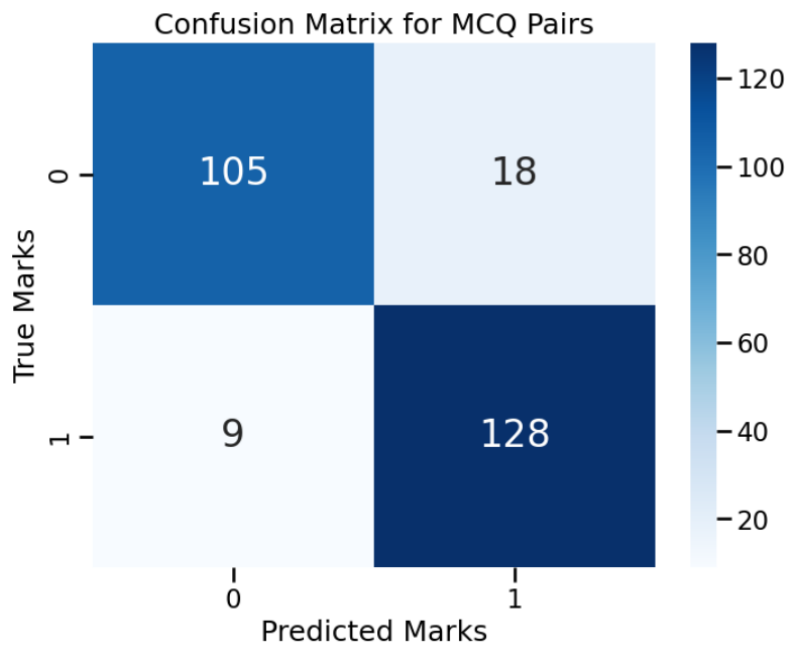


Figure 81: Confusion Matrix for MCQ mark assignment

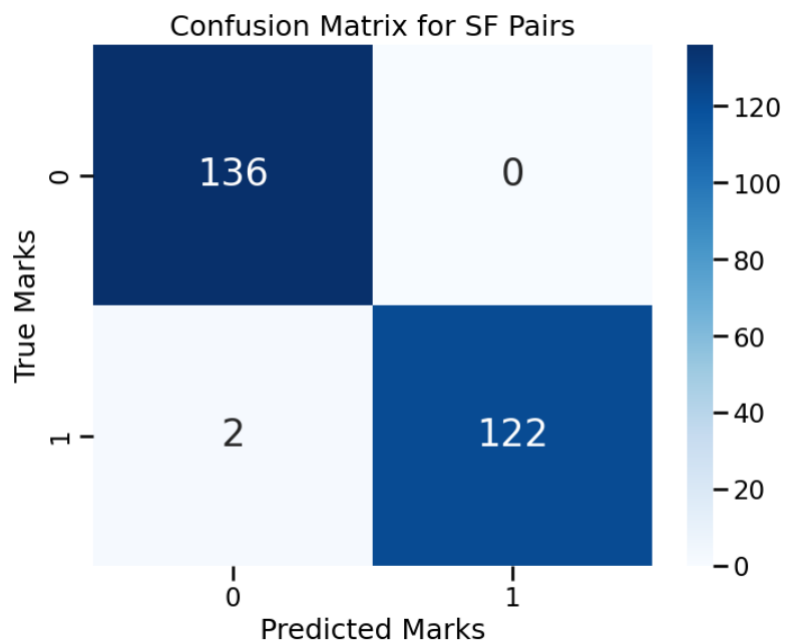


Figure 82: Confusion Matrix for SF mark assignment

Accuracy: 0.8961538461538462
Recall: 0.8961538461538462
F1-score: 0.8958333476694617
Precision: 0.8976887027896401

Figure 83: Classification Metrics for MCQ marking

Accuracy: 0.9923076923076923
Recall: 0.9923076923076923
F1-score: 0.9923044968798929
Precision: 0.9924191750278707

Figure 84: Classification Metrics for SF Marking

In this we can see a nearly perfect score for SF marking. This is likely due to the fact that there was little ambiguity in the parsing and marking, with even the possibility of an illegible or poorly spelled answer likely being fixed by chat-gpt’s tendency to correct spelling mistakes when processing inputs (Whittaker and Kitagishi, 2024).

MCQs seemed to perform a little worse. There seemed to be a positive skew in classification of 3.46%, but it was not deemed statistically significant.

question_no	question_no	question_text	question_options	student_answer	student_answer	mark_assigned	mark_earned	mark_justified	mark_schen	mark_schen	max_mark_s	max_mark_s	type	q_id
455	2.5.a	2.5.a	[What is it called? What is it called? B	A. cooling	1	0	0	0	B	B	1	1	1	3928
456	2.a	2.a	[Etikettes speed Etikettes speed D	A. genetic	1	1	1	1	Selected bocha D	D	1	1	1	1335
457	1.c	1.c	[Van der waals f Van der waals to B	A. atoms	0	1	1	1	Selected option D	D	1	1	1	10048
458	2.5.a	2.5.a	[What is it called? What is it called? D	B. grounding	0	1	1	1	[The student's a B	B	1	1	1	3928
459	2.a	2.a	[Etikettes speed Etikettes speed B	D. biochemical	0	1	1	1	[The student p a D	D	1	1	1	1335
461	1.c	1.c	[Van der waals f Van der waals to D	D. dipoles	1	1	1	1	[The student col D	D	1	1	1	10048
471	2.5.a	2.5.a	[What is it called? What is it called? D	A. cooling	0	1	1	1	Selected group B	B	1	1	1	3928
472	2.a	2.a	[Etikettes speed Etikettes speed C	B. grounding	0	0	0	0	D	D	1	1	1	1335
473	1.c	1.c	[Van der waals f Van der waals to D	C. charging	1	1	1	1	Selected D?	D	1	1	1	10048
479	2.5.a	2.5.a	[What is it called? What is it called? B	A. cooling	1	0	0	0	B	B	1	1	1	3928
480	2.a	2.a	[Etikettes speed Etikettes speed D	A. genetic	1	1	1	1	Selected correct D	D	1	1	1	1335
481	1.c	1.c	[Van der waals f Van der waals to A	A. atoms	0	0	0	0	D	D	1	1	1	10048
558	3.5.i	3.5.i	[In physics, what in physics, what C	A. machines	0	1	1	1	Selected option A	A	1	1	1	2022
565	3.5.i	3.5.i	[In physics, what in physics, what A	A. machines	1	1	1	1	[The student col A	A	1	1	1	2022
547	3.a.ii	3.a.ii	[In a longitudinal in a longitudinal B	D. magnitude	0	0	0	0	[The answer D: A	A	1	1	1	4247
558	3.5.i	3.5.i	[In physics, what in physics, what D	A. machines	0	0	0	0	A	A	1	1	1	2022
559	3.a.ii	3.a.ii	[In a longitudinal in a longitudinal D	A. amplitude	0	1	1	1	Selected option A	A	1	1	1	4247
616	2	2	[The head, thorax The head, thorax A	H	0	0	0	0	[The student's a D	D	1	1	1	4992
645	3.5	3.5	[What are the or What are the end D		0	0	0	0	[The student del B	B	1	1	1	9261
667	3.a	3.a	[What is the name What is the name D		1	0	0	0	[The student del D	D	1	1	1	5720

Figure 85: An example in which the entire question list was parsed instead of the student answer within our results

We also found that when looking at parsed student answers falling outside of “A”, “B”, “C” and “D”, the majority of false positives seemed to arise from the model segmenting the options lists which contained the final answer, and subsequently was awarded a mark due to the presence of the required answer.

Furthermore, there seemed to be a case in which an incorrect letter (“H” from the letter “A”) was parsed. Due to the effects of cropping and placing unpredictably-sized Calligrapher-generated handwriting, compounded by the problems caused in next-stroke predictions when generating single letters, it is almost certain the effects were the result of Exam Generation.

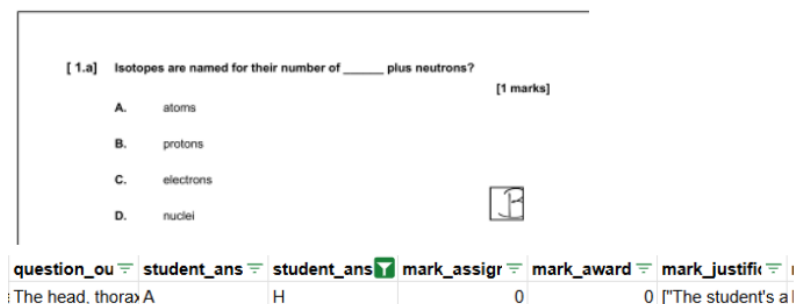


Figure 86: An example of the answer letter “B” being cropped due to Calligrapher pre-processing

The overall error rate was 10.38%, which is significantly lower than the one of 18% seen in our initial parser experiment. 3.076% of the error rate could be accounted for by segmenting errors and the rest as incorrect character parsing. While the possibility of parsing and marking errors deriving from hallucination remains ever-present due to the inherent LLM architecture properties (Yao et al., 2023), the majority of the error cases seen here could be eliminated through more tailored QPP prompting and the mitigation of data generation bugs.

During this experiment, we tested the software using supervised and unsupervised mark scheme setups and gained invaluable insights into the Evaluator’s mark scheme application ability. At first the mark scheme was presented to the model as is, with the model being expected to extrapolate mark weights and remove ambiguity from the at-times fractured mark schemes.

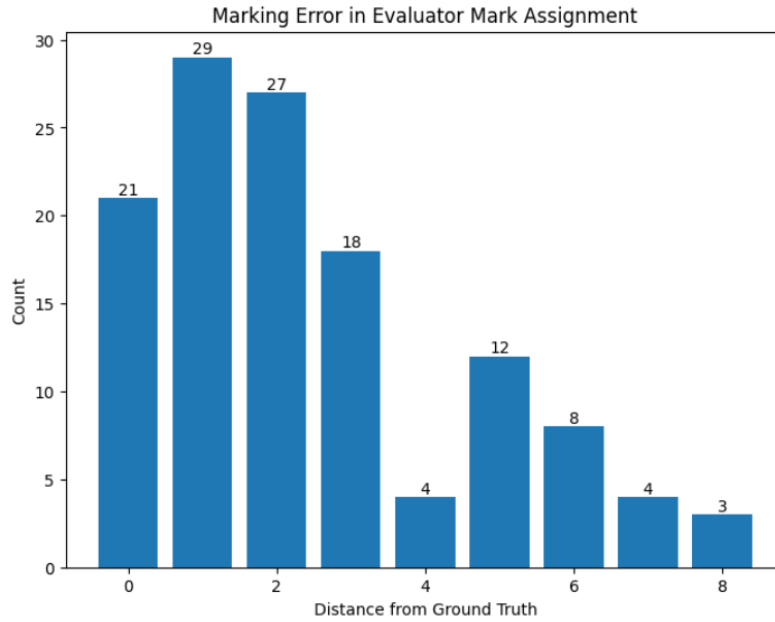


Figure 87: Distance between the ground truth awarded mark and the one predicted by the Evaluator

The raw mark schemes might have been sufficient for teachers with deep knowledge of the subject and answering habits of students but proved to not display a very concrete marking pattern between marking rounds, nor alignment with the distribution of the Ground Truth.

In the first part of the experiment we saw highly anomalous results, with 45.45% of marking regressions being grouped around 1–2 mark distance from GT. The results had a Mean Average Error of 2.48 and Mean Squared Error of 10.63. Though a 1–2 mark difference is relatively small, it would reinforce teacher distrust in the consistency in results expressed in the Charter School North Dulwich Survey, and would make the tool unacceptable for use within our use-case.

After supervised adaptation of the mark scheme (see **Section 4.2.2**) the results were improved drastically.

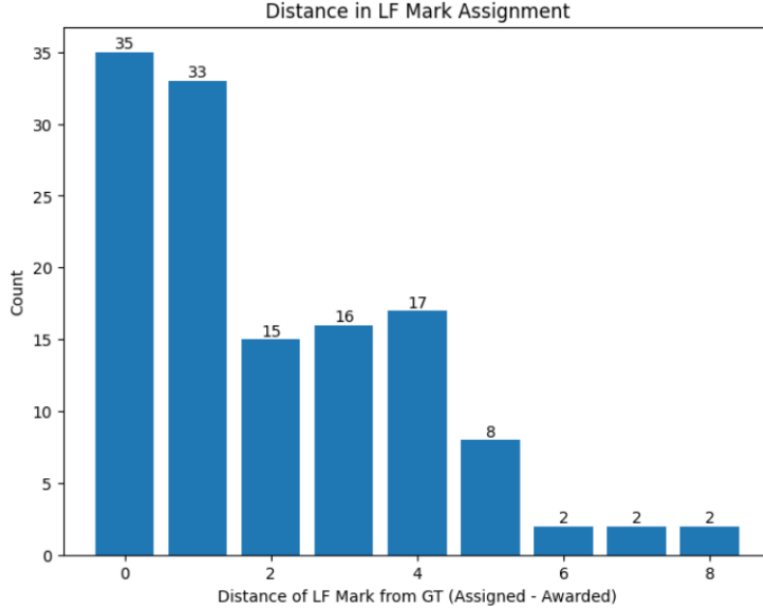


Figure 88: Distance between the ground truth awarded mark and the one given by the Evaluator with optimised mark scheme being presented to the model

We observe a 23.38% drop in MAE to 2.01 and a 37.71% drop in MSE to 7.75. The Mode now being the mark pertaining to perfect prediction by the evaluator entails that the model is extremely capable in reasoning on the presence and use of certain concepts, and still being able to handle a good amount of discretion (for example when distinguishing between a “partial” or “full explanation”).

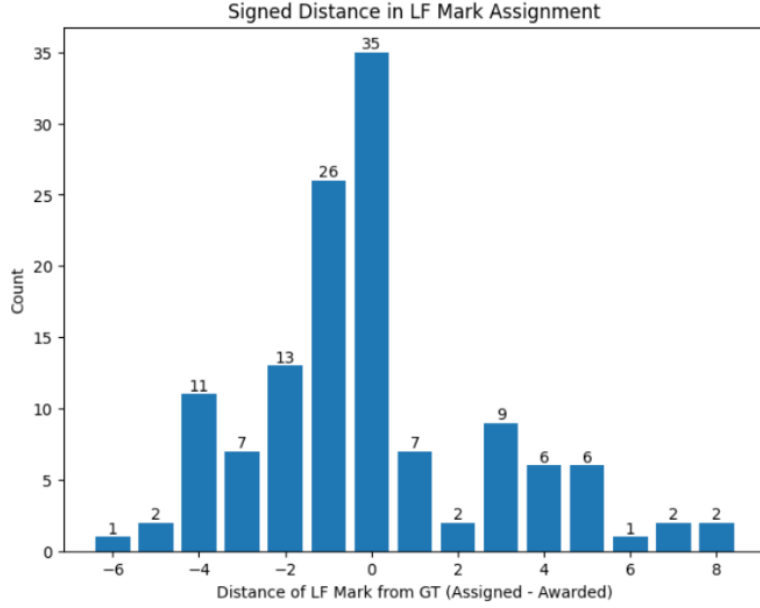


Figure 89: Signed distance of Mark Awarded from GT

When looking at marking bias it is evident that the evaluators marking had a noticeable lean towards the negative, with 371% more results skewed by -1 than +1. The most reasonable explanation for this would be the presence of an open ended mark being awarded/removed for the presence of an “incorrect statement” being included within the answer.

Such open-ended marks would for human teachers be a useful way to adjust grading by leveraging their knowledge of the curriculum the test is based upon and spotting statements overtly contradictory to the taught content. For our use case, such a mark is likely to be triggered by the evaluator due to the tendency for LLMs to be highly sensitive to sentiment-inverting phrases (**Yamamura and Ganguli, 2024**). Furthermore, the nature of the “incorrect statement” is completely ambiguous and not adjustable by supervised mark scheme adjustment due to insufficient information within our dataset.

It is also important to note we omitted a -0.5 mark allocation for a “calculation error” as we knew the data was not available to us and would exacerbate this bias.

7.69% of marked LFs were still awarded 0 marks with no justification despite an answer being parsed. Exactly half of these were due to incorrect concatenation of Text Block overflows, but with the other half being well formed

answers with some even gaining near perfect marks. This does confirm some of the concerns with inherent unpredictability of using LLMs for marking and warrants further research into mitigation of such cases, either by remarking, integration of fall-back human marking or other yet unexplored solutions.

To find the overall performance, we found the distance of the total mark in the GT test and its marked counterpart and normalised it to the total marks available in the test. The average Normalised Distance of GT from Prediction was 0.1375, meaning that on average each test was 13.75% of its total mark away from the GT value.

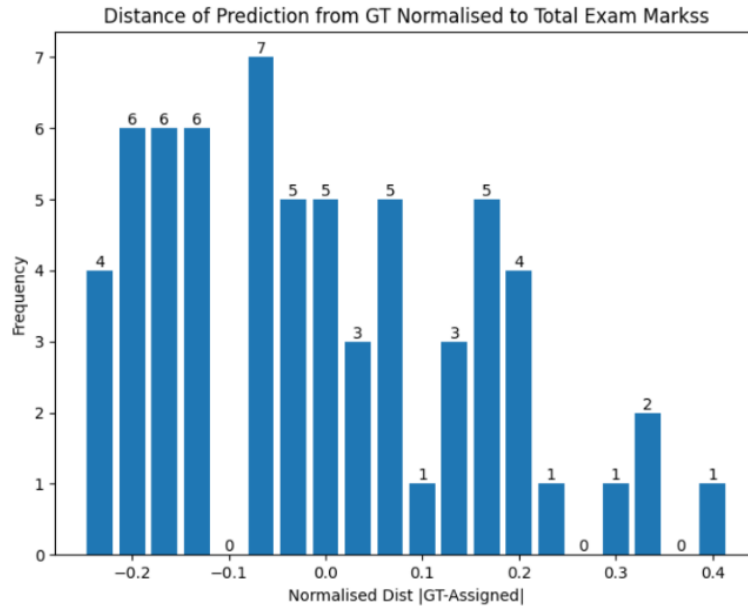


Figure 90: Normalised Distance of Prediction from GT in Total Marks Awarded

Between SFs and LFs there were no cases in which a mark was awarded with no justification.

9 Discussion

9.1 Findings

In this experiment we can clearly see that even with an early-stage prototype we have an extremely strong foundation for a new paradigm in GCSE exam

marking methodology.

According to Ofqual (**Keating, 2019**)(**Ofqual, 2013**), the probability of receiving a principal grade in mathematics was 96%, while within subjects with longer form answers such as in English this percentage decreases to 55–60%. The Normalised Mean Prediction distance of 13.5% indicates that the model, while not currently robust enough to be deployed within institutions, is extremely powerful in creating marking outcomes that are at least comparable to the success of teachers in replicating the principal mark.

Within LFs, where reasoning models were applied, the Evaluator was in complete agreement with the Examiner grade 26.9% of the time, and within 1 mark of the ground truth 61.05% of the time. As shown with SF questions the model was able to achieve an overall accuracy of 99.2%.

We also find a number of shortfalls: image encoding artifact disruptions, segmentation and MCQ OCR recognition errors when parsing, a potential positive bias in classification, the need for supervision when adapting mark schemes for marking, and bias in cases of ambiguity.

Fortunately the frequency of these errors was few and, aside from the artifact errors, we can attribute these failures to the dataset generation algorithm, which can be improved in future iterations. Segmentation errors could be readily solved with more tailored prompts, with more targeted mentions of layout elements. The MCQ errors would be significantly less (if at all) prevalent if character cutoffs and line interceptions were solved. The marking bias should be investigated but does not appear to be statistically significant, effects would have to be measured across datasets.

9.2 Future Research

The key insight from our project is undoubtedly that, given correct supervision, reasoning models are undoubtedly capable of carrying out effective criteria based evaluation of answers.

This supervision could take the form of the exam board providing mark-schemes that could be easier interpreted by LLM reasoning mechanisms as a standard, ensuring QPP prompts are robust for any given examination, or simply having the teachers adapt them to their standards during mock season.

All of these approaches would require re-thinking what digital skills should be training requirements for professionals within teaching, which are already crucial to adapting educational institutions for developments within AI (**Ng et al., 2023**).

Further experimentation is required in increasing this software’s cost-effectiveness. Experimentation with models of various sizes should be tested for cost-performance trade-off, as smaller and even non-LLM solutions could be promising in the place of the 4o-mini model.

For institutions or use-cases with predictably static exam paper structures where high zero-shot or few-shot performance on question paper parsing is not needed, a bespoke MSP solution should be implemented. Doing so has the potential to cut costs by up to 60% (**see Section 5.4**) for any one user but would have to be handled on a by-user basis.

Another aspect yet to be explored would be a per-institution fine-tuned local LLM integration. This has unprecedented potential for cutting costs and improving performance, as the models we have used so far have been extremely general and been called through public APIs, which could be a security risk for many institutions (**Kumar and Ahmed, 2024**).

Lastly, the evaluation process itself has many ways it could be made robust given cost is not an issue. Models such as gpt-o1, which boasts significantly better performance but a per-token cost almost x100 times higher (**PromptHub, 2024b**)(**OpenAI, 2025**) as well as the previously mentioned Ensemble ToT (**Ito and Ma, 2025**) evaluation approach could be used in order to provide even more robust performance.

9.3 Limitations

There are some limitations with our implementation, the most prominent one being our datasets. Both of our data sources are tied to the domain of science, which have been shown to be domain reasoning models that are most performant in (**see Section 6.1**). Future study should focus on finding data within domains such as Mathematics or English where more varied visual elements such as equations or longer answer lengths could be found.

10 Conclusion

In conclusion, we have succeeded in our goal in deploying an AEM pipeline MVP that delivers marking results that, with the named optimisations, have the potential to be the same or better when compared to real teachers.

The solution we created has a Normalised Mean Mark Prediction Distance of 0.1375, with a marking being done at fraction of the time and cost, by ensuring to take into account the bottlenecks in the deployment such a system, optimising for compute and implementing cheaper Parsing/Evaluation methods where possible.

We have also succeeded in developing a synthetic test standard for the Exam Board QAQ, where we were able to extract numerous insights for how adaptations for AI in examination papers might look like. We found that supervision when developing a mark scheme to be LLM-ready was necessary, but could be isolated to the Exam Board level and distributed to institutions, with the process of the adaptation to new exam standards/layouts being extremely uncostly.

Numerous future improvements for users/researchers looking to implement this project were suggested, whether looking to optimise for marking accuracy, speed, compute or security. Considerations were also made on how the solution could be scaled to the institutional level. These optimisations no doubt needed to be considered before AEM pipeline application is used within real schools, but we have shown that it might now be time for institutions to begin an earnest effort to look into widespread AI training programmes within teaching professionals, as well as meaningful AI integration into core examination systems.

References

Green, F., Felstead, A., Gallie, D. and Henseke, G. (2018) Work intensity in Britain: First findings from the Skills and Employment Survey 2017. [Project Report] London: Centre for Learning and Life Chances in Knowledge Economies and Societies, UCL Institute of Education. doi: <https://orca.cardiff.ac.uk/id/eprint/115439/>. Accessed: 10 February 2025.

Sonkar, S., Ni, K., Lu, L.T., Kincaid, K., Hutchinson, J.S., and Baraniuk, R.G. (2024) Automated Long Answer Grading with RiceChem Dataset. arXiv preprint arXiv:2404.14316. Available at: <https://arxiv.org/abs/2404.14316>. Accessed: 11 February April 2025.

Zhong, Q., Ding, L., Liu, J., Du, B. and Tao, D. (2023) Can ChatGPT Understand Too? A Comparative Study on ChatGPT and Fine-tuned BERT. arXiv preprint arXiv:2302.10198v2 [cs.CL]. doi: <https://arxiv.org/abs/2302.10198>. Accessed: 11 February 2025.

Ito, Y. and Ma, Q. (2025) Ensemble ToT of LLMs and Its Application to Automatic Grading System for Supporting Self-Learning. Preprint. Kyoto University & Kyoto Institute of Technology, 25 February. doi: <https://arxiv.org/abs/2502.16399v1> Accessed: 13 Feb 2025

Morris, R., Gorard, S., See, B.H. and Siddiqui, N. (2024) Can a code-based approach to marking and feedback reduce teachers' workload? An evaluation of the FLASH marking intervention. *Oxford Review of Education*, 50(4), pp.552–569. doi: <https://doi.org/10.1080/03054985.2023.2258779>. Accessed: 13 February 2025.

Green, F., Felstead, A., Gallie, D. and Henseke, G. (2018) Work Intensity in Britain: First Findings from the Skills and Employment Survey 2017. London: Centre for Learning and Life Chances in Knowledge Economies and Societies (LLAKES), UCL Institute of Education. doi: <https://orca.cardiff.ac.uk/id/eprint/115439/>. Accessed: 13 February 2025.

Henderson-Brooks, C. (2021) Marking as emotional labour: A discussion of the affective impact of assessment feedback on enabling educators. *International Studies in Widening Participation*, 8(1), pp. 110–121. doi: <https://nova.ocs.newcastle.edu.au/ceehe/index.php/iswp/article/view/158>. Accessed: 13 February 2025.

Fischer, I. (2024) Evaluating the ethics of machines assessing humans. *Journal of Information Technology Teaching Cases*, 14(2), pp.273–281. doi: <https://doi.org/10.1177/20438869231178844>. Accessed: 13 February 2025.

Dupriez, V., Delvaux, B. and Lothaire, S. (2016) Teacher shortage and attrition: Why do they leave? *British Educational Research Journal*, 42(1), pp.21–39. doi: <https://doi.org/10.1002/berj.3193>. Accessed: 13 February

2025.

Department for Education and The Rt Hon Gillian Keegan. (2023) Record number of teachers in England's schools. GOV.UK. Available at: <https://www.gov.uk/government/news/record-number-of-teachers-in-englands-schools> (Accessed: 16 Feb 2025).

Department for Education. (2024) Schools, pupils and their characteristics: Academic year 2023/24. GOV.UK. Available at: <https://explore-education-statistics.service.gov.uk/find-statistics/school-pupils-and-their-characteristics/2023-24> (Accessed: 16 Feb 2025). Explore our statistics and data

Full Fact. (2023), England has fewer teachers relative to the number of school pupils than in 2010. Full Fact. Available at: <https://fullfact.org/education/teacher-pupil-numbers/> (Accessed: 16 Feb 2025).

Bergviken Rensfeldt, A. and Rahm, L. (2023) Automating teacher work? A history of the politics of automation and artificial intelligence in education. *Postdigital Science and Education*, 5, pp. 25–43. doi: <https://doi.org/10.1007/s42438-022-00344-x>. Accessed: 16 Feb 2025.

UK Government. (2018) Data Protection Act 2018, Schedule 2, Part 4: Exam scripts and exam marks. [legislation.gov.uk](https://www.legislation.gov.uk/ukpga/2018/12/schedule/2/part/4/crossheading/exam-scripts-and-exam-marks). Available at: <https://www.legislation.gov.uk/ukpga/2018/12/schedule/2/part/4/crossheading/exam-scripts-and-exam-marks> (Accessed: 16 February 2025).

Kumar, B.V.P. and Ahmed, M.D.S. (2024) Beyond Clouds: Locally Runnable LLMs as a Secure Solution for AI Applications. *Digital Society*, 3(49). doi: <https://doi.org/10.1007/s44206-024-00141-y>. Accessed: 16 February 2025.

Sun, W., Wang, J., Guo, Q., Li, Z., Wang, W. and Hai, R. (2024) CEBench: A Benchmarking Toolkit for the Cost-Effectiveness of LLM Pipelines. *arXiv preprint arXiv:2407.12797v1 [cs.PF]*. doi: <https://arxiv.org/abs/2407.12797v1>. Accessed: 14 February 2025.

Yarlagadda, R.T. (2018) Python Engineering Automation to Advance Artificial Intelligence and Machine Learning Systems. *International Journal of Innovations in Engineering Research and Technology (IJIERT)*, 5(6), pp.87–92. ISSN: 2394-3696. doi:<https://link.springer.com/article/10.1007/s44206-024-00141-y> Accessed: 28 February 2025.

Díaz, M., Mendoza-García, A., Ferrer, M.A. and Sabourin, R. (2025) A survey of handwriting synthesis from 2019 to 2024: A comprehensive review. *Pattern Recognition*, 162, 111357. doi: <https://doi.org/10.1016/j.patcog.2025.111357>. Accessed: 28 February 2025.

Carter, S., Ha, D., Johnson, I., and Olah, C. (2016) Experiments in Handwriting with a Neural Network. *Distill*. doi: 10.23915/distill.00004. Available at: <https://distill.pub/2016/handwriting/>. Accessed: 28 April 2025.

Daware, S. (2021) Text-to-Handwriting. GitHub repository. Available at: <https://github.com/saurabhdaware/text-to-handwriting>. Accessed: 10 April 2025.

Graves, A. (2013) Generating Sequences With Recurrent Neural Networks. arXiv preprint arXiv:1308.0850. doi: <https://arxiv.org/abs/1308.0850>. Accessed: 28 February 2025.

Du, M., Chen, Y., Wang, Z., Nie, L. and Zhang, D. (2024) LLM4ED: Large Language Models for Automatic Equation Discovery. *Physics of Fluids*, 36(9), 097121. doi: <https://doi.org/10.1063/5.0209740>. Accessed: 2 March February 2025.

OpenAI. (2024) OpenAI o1 System Card. Available at: <https://openai.com/index/openai-o1-system-card/>. Accessed: 2 March 2025.

Li, C., Liang, J., Zeng, A., Chen, X., Hausman, K., Sadigh, D., Levine, S., Fei-Fei, L., Xia, F., and Ichter, B. (2023) Chain of Code: Reasoning with a Language Model-Augmented Code Emulator. arXiv preprint arXiv:2312.04474v3. Available at: <https://arxiv.org/abs/2312.04474v3>. Accessed: 16 March 2025.

PromptHub. (2024) GPT-4o mini Model Card. Available at: <https://www.prompthub.us/models/gpt-4o-mini>. Accessed: 16 March 2025.

PromptHub. (2024) o1-mini Model Card. Available at: <https://www.prompthub.us/models/o1-mini>. Accessed: 16 March 2025.

Zheng, M., Pei, J., Logeswaran, L., Lee, M., and Jurgens, D. (2024) When "A Helpful Assistant" Is Not Really Helpful: Personas in System Prompts Do Not Improve Performances of Large Language Models. arXiv preprint arXiv:2311.10054v3. doi: <https://doi.org/10.48550/arXiv.2311.10054>. Accessed: 16 March 2025.

Chen, Z., Li, J., Chen, P., Li, Z., Sun, K., Luo, Y., Mao, Q., Yang, D., Sun, H., and Yu, P.S. (2025) Harnessing Multiple Large Language Models: A Survey on LLM Ensemble. arXiv preprint arXiv:2502.18036. doi: <https://doi.org/10.48550/arXiv.2502.18036>. Accessed: 16 March 2025.

Rowtula, V., Oota, S.R., and Jawahar, C.V. (2019) Towards Automated Evaluation of Handwritten Assessments. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, pp. 426–433. doi: <https://doi.org/10.1109/ICDAR.2019.00075>. Accessed: 16 March 2025.

Rodrigues, G.A.P., Serrano, A.L.M., Vergara, G.F., Albuquerque, R.d.O., and Nze, G.D.A. (2024) Impact, Compliance, and Countermeasures in Relation to Data Breaches in Publicly Traded U.S. Companies. *Future Internet*, 16(6), 201. doi: <https://doi.org/10.3390/fi16060201>. Accessed: 17 March 2025

Sherstinsky, A. (2020) Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306. doi: <https://doi.org/10.1016/j.physd.2019.132306>. Accessed: 17 March 2025.

Qin, C., Chen, L., Cai, Z., Liu, M. and Jin, L. (2023) Long short-term memory with activation on gradient. *Neural Networks*, 164, pp.135–145. doi: <https://doi.org/10.1016/j.neunet.2023.04.026>. Accessed: 17 March 2025.

Ribeiro, A.H., Tiels, K., Aguirre, L.A., and Schön, T.B. (2020) Beyond Exploding and Vanishing Gradients: Analysing RNN Training Using Attractors and Smoothness. arXiv preprint arXiv:1906.08482. doi: <https://doi.org/10.48550/arXiv.1906.08482>. Accessed: 10 April 2025.

Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018) Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, 2018, 7068349. doi: <https://doi.org/10.1155/2018/7068349>. Accessed: 18 March 2025.

Mulfari, D., Celesti, A., Fazio, M., Villari, M. and Puliafito, A. (2016) Using Google Cloud Vision in Assistive Technology Scenarios. In: *IEEE Workshop on ICT Solutions for eHealth 2016*. IEEE. doi: UNKNOWN. Accessed: 18 March 2025.

Chai, D. (2016) Automated Marking of Printed Multiple Choice Answer

Sheets. In: 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Bangkok, Thailand, 7–9 December. IEEE, pp. 145–149. doi: <https://ieeexplore.ieee.org/abstract/document/7851785>. Accessed: 18 March 2025.

Bluche, T., Louradour, J. and Messina, R. (2017) Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November. IEEE, pp. 1050–1055. doi: <https://doi.org/10.1109/ICDAR.2017.174>. Accessed: 18 March 2025.

Besta, M., Barth, J., Schreiber, E., Kubicek, A., Catarino, A., Gerstenberger, R., Nyczyk, P., Iff, P., Li, Y., Houliston, S., Sternal, T., Copik, M., Kwaśniewski, G., Müller, J., Flis, L., Eberhard, H., Niewiadomski, H. and Hoefler, T. (2025) Reasoning Language Models: A Blueprint. arXiv preprint arXiv:2501.11223v1. doi: <https://doi.org/10.48550/arXiv.2501.11223>. Accessed: 20 March 2025.

OpenAI. (2025) API Pricing. Available at: <https://openai.com/api/pricing/>. Accessed: 20 March 2025.

Keating, K. (2019) '11 things we know about marking and 2 things we don't ...yet'. The Ofqual Blog, 5 March. Available at: <https://ofqual.blog.gov.uk/2019/03/05/14572/>. Accessed: 20 March 2025.

Ofqual (2013) GCSE Reform Consultation June 2013. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/529371/2013-06-11-gcse-reform-consultation-june-2013.pdf. Accessed: 20 March.

Alkafaween, U. (2025) 'Automating Autograding: Large Language Models as Test Suite Generators for Introductory Programming'. Journal of Computer Assisted Learning. Available at: <https://onlinelibrary.wiley.com/doi/full/10.1111/jcal.13100>. Accessed: 20 March 2025.

Chu, Y., He, P., Li, H., Han, H., Yang, K., Xue, Y., Li, T., Krajcik, J. and Tang, J. (2025) Enhancing LLM-Based Short Answer Grading with Retrieval-Augmented Generation. arXiv preprint arXiv:2504.05276. Available at: <https://arxiv.org/abs/2504.05276>. Accessed: 20 March 2025.

Sam, D., Finzi, M. and Kolter, J.Z. (2025) Predicting the Performance

of Black-box LLMs through Self-Queries. arXiv preprint arXiv:2501.01558. Available at: <https://arxiv.org/abs/2501.01558>. Accessed: 22 March 2025.

Fan, Z., Wang, W., Wu, X. and Zhang, D. (2025) SedarEval: Automated Evaluation using Self-Adaptive Rubrics. arXiv preprint arXiv:2501.15595. Available at: <https://arxiv.org/abs/2501.15595>. Accessed: 22 March 2025.

BCS, The Chartered Institute for IT (2022) BCS Code of Conduct for Members. Version 8, reviewed and approved by Trustee Board on 8 June. Available at: <https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf>. Accessed: 22 March 2025.

Demszky, D., Liu, J., Hill, H.C., Sanghi, S., and Chung, A. (2025) Automated feedback improves teachers' questioning quality in brick-and-mortar classrooms: Opportunities for further enhancement. *Computers & Education*, 227, 105183. doi: <https://doi.org/10.1016/j.compedu.2024.105183>. Accessed: 22 March 2025.

Fowler, J. (2024) Title of the Thesis. Ph.D. thesis, Durham University. Available at: <http://etheses.dur.ac.uk/15238/1/Fowler-000267271.pdf>. Accessed: 23 March 2025.

Senarath, U.S. (2021) Waterfall Methodology, Prototyping and Agile Development. Technical Report.

Cramer, T., Friedman, R., Miller, T., Seberger, D., Wilson, R. and Wolczko, M. (1997) Compiling Java Just in Time. *IEEE Micro*, 17(3), pp. 36–53. doi: <https://doi.org/10.1109/40.591653>. Accessed: 23 March 2025.

Lenarduzzi, V. and Taibi, D. (2016) MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product. In: 42nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 31 August–2 September, Limassol, Cyprus. IEEE, pp. 112–119. doi: <https://doi.org/10.1109/SEAA.2016.56>. Accessed: 23 March 2025.

Tudor, D. and Walter, G.A. (2006) Using an Agile Approach in a Large, Traditional Organization. In: *Proceedings of AGILE 2006 Conference (AGILE'06)*. IEEE, pp. 1–10. doi: <https://doi.org/10.1109/AGILE.2006.34>. Accessed: 23 March 2025

Willman, J. (2021) 'Overview of PyQt5', in *Modern PyQt*. Apress, Berke-

ley, CA, pp. 1–42. doi: https://doi.org/10.1007/978-1-4842-6603-8_1. Accessed: 24 March 2025.

Auger, T. and Saroyan, E. (2024) 'Overview of the OpenAI APIs', in *Generative AI for Web Development*. Apress, Berkeley, CA, pp. 87–116. doi: https://doi.org/10.1007/979-8-8688-0885-2_6. Accessed: 24 March 2025.

Shekhar, S., Dubey, T., Mukherjee, K., Saxena, A., Tyagi, A., and Kotla, N. (2024) Towards Optimizing the Costs of LLM Usage. arXiv preprint arXiv:2402.01742. Available at: <https://arxiv.org/abs/2402.01742>. Accessed: 24 March 2025.

Rodrigues, G.A.P., Serrano, A.L.M., Vergara, G.F., Albuquerque, R.d.O., and Nze, G.D.A. (2024) 'Impact, Compliance, and Countermeasures in Relation to Data Breaches in Publicly Traded U.S. Companies'. *Future Internet*, 16(6), 201. doi: <https://doi.org/10.3390/fi16060201>. Accessed: 24 March 2025.

Al-Selwi, S.M., Hassan, M.F., Abdulkadir, S.J., Muneer, A., Sumiea, E.H., Alqushaibi, A., and Ragab, M.G. (2024) RNN-LSTM: From applications to modeling techniques and beyond—Systematic review. *Journal of King Saud University - Computer and Information Sciences*, 36, 102068. doi: <https://doi.org/10.1016/j.jksuci.2024.102068>. Accessed: 27 March 2025.

Asharabi, R.M. and Alhazmi, S.M. (2024) Accelerating the convergence of a two-dimensional periodic nonuniform sampling series through the incorporation of a bivariate Gaussian multiplier. *AIMS Mathematics*, 9(11), pp. 30898–30921. doi: <https://doi.org/10.3934/math.20241491>. Accessed: 27 March 2025.

Geng, Y., Li, Q., Yang, G., and Qiu, W. (2024) Logistic Regression. In: *Practical Machine Learning Illustrated with KNIME*. Springer, Singapore, pp. 99–132. doi: https://doi.org/10.1007/978-981-97-3954-7_4. Accessed: 27 March 2025.

Zhang, Q., Huang, V.S.-J., Wang, B., Zhang, J., Wang, Z., Liang, H., Wang, S., Lin, M., He, C., and Zhang, W. (2024) Document Parsing Unveiled: Techniques, Challenges, and Prospects for Structured Information Extraction. arXiv preprint arXiv:2410.21169. Available at: <https://arxiv.org/abs/2410.21169>. Accessed: 28 March 2025.

Paretta, R.L. and Chadwick, L.W. (1975) The Sequencing of Examination Questions and Its Effects on Student Performance. *The Accounting Review*, 50(3), pp. 595–601. Available at: <https://www.jstor.org/stable/245020>. Accessed: 28 March 2025.

Spalding, V. (2010) Structuring and Formatting Examination Papers: Examiners’ Views of Good Practice. Centre for Education Research and Policy. Available at: <https://filestore.aqa.org.uk/content/research/CERP-RP-VS-05022010-0.pdf>. Accessed: 2 April 2025.

Bangalore, S. and Joshi, A.K. (1999) Supertagging: An Approach to Almost Parsing. *Computational Linguistics*, 25(2), pp. 237–265. Available at: <https://aclanthology.org/J99-2004.pdf>. Accessed: 2 April 2025.

Yamamura, A. and Ganguli, S. (2024) Fooling LLM Graders into Giving Better Grades through Neural Activity Guided Adversarial Prompting. *arXiv preprint arXiv:2412.15275*. Available at: <https://arxiv.org/abs/2412.15275>. Accessed: 2 April 2025.

Ferraioli, D., Penna, P., and Ventre, C. (2022) Two-Way Greedy: Algorithms for Imperfect Rationality. In: Feldman, M., Fu, H., and Talgam-Cohen, I. (eds) *Web and Internet Economics. WINE 2021. Lecture Notes in Computer Science*, vol 13112. Springer, Cham, pp. 3–21. doi: https://doi.org/10.1007/978-3-030-94676-0_1. Accessed: 2 April 2025.

Dolfi, M., Mieskes, M., and Wiegand, M. (2022) DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’22)*, Washington, D.C., USA. ACM, pp. 4596–4606. doi: <https://doi.org/10.1145/3534678.3539043>. Accessed: 2 April 2025.

Wedholm, W. (2024) Exploring the Influence of Data Formats on the Consistency of Large Language Models Outputs. Bachelor’s thesis, Mälardalen University, School of Innovation, Design and Engineering. Available at: <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1868676>. Accessed: 2 April 2025.

Shahriar, S., Lund, B.D., Mannuru, N.R., Arshad, M.A., Hayawi, K., Bevara, R.V.K., Mannuru, A., and Batool, L. (2024) Putting GPT-4o to the Sword: A Comprehensive Evaluation of Language, Vision, Speech, and Multi-modal Proficiency. *Applied Sciences*, 14(17), 7782. doi: <https://doi.org/10.3390/app14177782>.

Accessed: 4 April 2025.

Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., He, Y., and Xue, H. (2022) Towards Robust Vision Transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June. IEEE, pp. 12042–12051. doi: <https://doi.org/10.1109/CVPR52688.2022.01173>. Accessed: 4 April 2025.

Liu, Y., Hu, J., Shan, Y., Li, G., Zou, Y., Dong, Y., and Xie, T. (2025) LLMigrate: Transforming “Lazy” Large Language Models into Efficient Source Code Migrators. arXiv preprint arXiv:2503.23791. Available at: <https://arxiv.org/abs/2503.23791>. Accessed: 4 April 2025.

Liu, Y., Niranjana, M., and Jacques, R. (2006) Double Regularization for Unifying Discriminative and Generative Learning. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006), Hong Kong, China, 20–24 August. IEEE, pp. 568–571. doi: <https://doi.org/10.1109/ICPR.2006.1017>. Accessed: 4 April 2025.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q.V., and Zhou, D. (2022) Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: Advances in Neural Information Processing Systems 35 (NeurIPS 2022).

Hu, T. and Collier, N. (2024) Quantifying the Persona Effect in LLM Simulations. arXiv preprint arXiv:2402.10811. Available at: <https://arxiv.org/abs/2402.10811>. Accessed: 4 April 2025.

Mervaala, E., and Kousa, I. (2025) Out of Context! Managing the Limitations of Context Windows in ChatGPT-4o Text Analyses. Journal of Data Mining & Digital Humanities, NLP4DH. doi: <https://doi.org/10.46298/jdmdh.15090>. Accessed: 4 April 2025.

Department for Education. (2024) Teacher Pay Data Set from School Workforce in England. Available at: <https://explore-education-statistics.service.gov.uk/data-catalogue/data-set/eb7d4270-59c8-4b87-9e36-7eee926bd395>. Accessed: 4 April 2025.

Material Design. (2025) Canonical Layouts: List-Detail. Available at: <https://m3.material.io/foundations/layout/canonical-layouts/list-detail>. Accessed: 4 April 2025.

Apple Inc. (2024) Split Views. In: Human Interface Guidelines. Available at: <https://developer.apple.com/design/human-interface-guidelines/split-views>. Accessed: 4 April 2025.

Whittaker, E. and Kitagishi, I. (2024) Large Language Models for Simultaneous Named Entity Extraction and Spelling Correction. arXiv preprint arXiv:2403.00528. Available at: <https://arxiv.org/abs/2403.00528>. Accessed: 4 April 2025.

Yao, J.-Y., Ning, K.-P., Liu, Z.-H., Ning, M.-N., Liu, Y.-Y., and Yuan, L. (2023) LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples. arXiv preprint arXiv:2310.01469. Available at: <https://arxiv.org/abs/2310.01469>. Accessed: 4 April 2025.

Ng, D.T.K., Leung, J.K.L., Su, J., Ng, R.C.W., and Chu, S.K.W. (2023) Teachers' AI Digital Competencies and Twenty-First Century Skills in the Post-Pandemic World. *Educational Technology Research and Development*, 71, 137–161. doi: <https://doi.org/10.1007/s11423-023-10203-6>. Accessed: 4 April 2025.

Vasquez, S. (2018) Handwriting Synthesis with RNNs. Available at: <https://github.com/sjvasquez/handwriting-synthesis>. Accessed: 4 April 2025.

Flodén, J. (2024) Grading exams using large language models: A comparison between human and AI grading of exams in higher education using ChatGPT. *British Educational Research Journal*, 00, pp.1–24. doi: <https://doi.org/10.1002/berj.4069>. Accessed: 5 March 2025.

Morjaria, L., Burns, L., Bracken, K., Levinson, A.J., Ngo, Q.N., Lee, M. and Sibbald, M. (2024) Examining the Efficacy of ChatGPT in Marking Short-Answer Assessments in an Undergraduate Medical Program. *International Medical Education*, 3, pp.32–43. doi: <https://doi.org/10.3390/ime3010004>. Accessed: 5 March 2025.

Welbl, J., Liu, N.F., and Gardner, M. (2017) Crowdsourcing Multiple Choice Science Questions. arXiv preprint arXiv:1707.06209. Available at: <https://arxiv.org/abs/1707.06209>. Accessed: 5 March 2025. [92] Ito, Y. and Ma, Q. (2025) Ensemble ToT of LLMs and Its Application to Automatic Grading System for Supporting Self-Learning. arXiv preprint arXiv:2502.16399. Available at: <https://arxiv.org/abs/2502.16399>. Accessed: 10 April 2025.

A Appendix

Author's Note

The word count presented does not count references or figure descriptions

The author acknowledges that the length of this report may exceed typical expectations. This is primarily due to the extensive analysis undertaken to evaluate the performance of the software. Prior to submission, the length of the report was discussed with the project supervisor, who confirmed that the level of detail presented is appropriate for the depth of the research conducted.

All participants must be able to contact the investigator and/or the supervisor after the investigation. They should be given contact details for both student and supervisor as part of the debriefing.

11. The evaluation was described in detail with all of the participants at the beginning of the session, and participants were fully debriefed at the end of the session. All participants were given the opportunity to ask questions at both the beginning and end of the session.
Participants must be provided with sufficient information prior to starting the session, and in the debriefing, to enable them to understand the nature of the investigation.

12. All the data collected from the participants is stored securely, and in an anonymous form.
All participant data (hard-copy and soft-copy) should be stored securely (i.e. locked filing cabinets for hard copy, password protected computer for electronic data), and in an anonymised form.

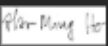
Project title: Automatic Exam Marking (AEM) Pipeline for GCSE-level Assessments

Student's Name: Jonas Papinigis
 Student's Registration Number: 22104766

Student's Signature: 

Date: 10/04/25

Supervisor's Name: Hsi-Ming Ho

Supervisor's Signature: 

Date: 10/04/2025

Version 3.1-3 -24/05/21

Figure 91: Screenshot of the user compliance form signed by supervisor and candidate